NOVA SCHOOL OF
SCIENCE & TECHNOLOGY

DEPARTAMENT OF MATERIAL SCIENCES

Tiago Nuno Gomes Dias

Bachelor of Science in Micro and Nanotechnologies Engineering

Development of an automated microfluidics-based device for simulating human gastrointestinal digestion

Master of Science in Micro and Nanotechnologies Engineering

Universidade NOVA de Lisboa

November, 2021

Development of an automated microfluidics-based device for simulating human gastrointestinal digestion

Tiago Nuno Gomes Dias
Bachelor of science in Micro and Nanotechnologies Engineering

Supervisor:	Miguel Xavier, Research Fellow, INL
Co-supervisor:	Hugo Águas, Associate Professor, NOVA University Lisbon

Judges:

Presidente:	Doutor Rodrigo Ferrão de Paiva Martins, Professor Catedrático do Departamento de Ciência dos Materiais da NOVA School of Science and Technology | FCT NOVA.
Arguentes:	Dr. Jonathan James West, Lecturer in Biomedical Microfluidics within Medicine at the University of Southampton

Master of Science in Micro and Nanotechnologies Engineering

Universidade NOVA de Lisboa
November, 2021

Development of an automated microfluidics-based device for simulating human gastrointestinal digestion

Acknowledgements

This thesis is a result of the project NORTE-01-0145-FEDER-000047, supported by Norte Portugal Regional Operational Programme (NORTE 2020), under the PORTUGAL 2020 Partnership Agreement, through the European Regional Development Fund (ERDF).

I would like to acknowledge the Science and Technology Faculty and to its teachers for providing an amazing campus and university experience, providing so many learning opportunities through these five years, being able to always help me when I wanted to know more about any subject and, most of all, for creating such an amazing and complete degree where I could learn about many of my areas of interest.

A big acknowledge must also be given to Miguel Xavier for receiving me and the whole of the Iberian Nanotechnology Laboratory (INL) as it was a great place to work in, with the best of people where I could learn more about any area of my interest. Besides thanking the whole of INL, I specifically thank Victor Calero for his help and companionship throughout my whole stay.

Finally, I must thank my family as they were always ready to help me in every situation and managed to pay for my education in a place away from home.

“If something’s important enough, you should try.
Even if the probable outcome is failure.”
(Elon Musk)

ABSTRACT

Understanding the fate of bioactive compounds under digestive conditions is critical for early-stage drug development. *In vitro* digestion methods offer an alternative to human and animal studies to reduce costs and complexity and avoid ethical concerns. Current *in vitro* digestion methods can be generally divided into the designations of static and dynamic models. While static models are typically simplified and lack the ability to fully reproduce the complexity of human digestion, dynamic models make use of large and complex equipment and require relatively large amounts of sample and reagents, making them unsuitable to study the digestion of valuable compounds, such as nano-delivery systems. In this work, a microfluidic-based method was developed to bridge the current gap between dynamic and static digestion models. The device was semi-automated and miniaturized to allow reducing sample and reagent usage, operator inputs, and downsize lab occupation. Digestion-chips were fabricated following a layer-by-layer approach from micro-milled and laser cut poly methyl methacrylate (PMMA) layers. Temperature and pH sensors were integrated to monitor key digestion parameters and worked in direct feedback with resistors for heating and syringe pumps injecting HCl or NaOH for real-time pH adjustments. Using this device, gastric and intestinal digestions were successfully replicated and validated using a casein reporter molecule against the gold standard static digestion protocol defined within the INFOGEST Cost action. In addition, the peristaltic pump developed within this work will allow to include important features of digestion such as gastric emptying, thus increasing the physiological relevance of the model. Overall, this work achieved a semi-automated and miniaturized device for human digestive simulations. The device was validated against gold standard protocols and showed the potential to be used in laboratory settings for *in vitro* testing of compounds aiming for oral administration using reduced volumes and with minimal operator input.

Keywords:
In vitro digestion models, Semi-dynamic digestion, Microfluidics, Sensors, automation

RESUMO

Compreender o trajeto de componentes bio ativos quando submetidos às condições do sistema digestivo humano é crítico para o desenvolvimento inicial de medicamentos administrados por via oral. Os métodos de digestão *in vitro* oferecem uma alternativa aos testes feitos em animais e humanos reduzindo assim o custo e a complexidade dos mesmos, bem como evitando questões éticas. Atualmente, os sistemas de digestão são divididos em dois grandes grupos, os modelos estáticos e os dinâmicos. Os primeiros são bastante simplificados e não são capazes de reproduzir com grande precisão o sistema de digestão humano. Os modelos dinâmicos, por outro lado, são capazes de o fazer mas requerem equipamento de elevadas dimensões e complexidade que utilizam grandes quantidades de amostras e reagentes o que é bastante prejudicial quando estas são de produção difícil tal como nano-componentes. Neste trabalho, foi desenvolvido um sistema microfluídico que faz a ponte entre os sistemas de digestão dinâmicos e estáticos atuais. O dispositivo é semi automatizado e de reduzidas dimensões sendo, assim, capaz de reduzir a quantidade de amostra e reagente necessários, o *input* de utilizadores, e a ocupação de espaço laboratorial. Os despositivos de digestão foram fabricados utilizando o método *layer-by-layer*, sendo as camadas utilizadas constituidas por polimetilmetacrilato (PMMA). Os sensores de temperatura e pH foram integrados para monitorizar estes parametros cruciais da digestão e funcionaram para um controlo de feedback com aquecimento por resistência bem como controlo da injeção de HCl e NaOH feito por *syringe pumps*. Este dispositivo, quando comparado ao protocolo INFOGEST replica, com sucesso, os fases digestivas gástrica e intestinal utilizando uma molécula de teste de caseína. Além disso, a *pump* peristáltica desenvolvida e testada é também capaz de replicar funções importantes como o *gastric emptying*, aumentando, assim, a relevância fisiológica do modelo. No geral, este trabalho conseguiu produzir uma simulação de digestão humana semi automatizada e miniaturizada que foi validada utilizando modelos standard de digestão estática. O modelo desmonstrou, assim, a capacidade de reproduzir digestões *in vitro* em laboratório para o teste de compostos para administração oral reduzindo os volumes utilizados e não requerendo grande *input* pelo utilizador.

Palavras-chaves: modelos de digestão *In vitro*, digestão semi dinâmica, microfluídica, sensores, automatização

INDEX

LIST OF ABREVIATIONS AND NOTATIONS

PDMS – Polydimethylsiloxane

PMMA – Poly (methyl methacrylate)

CNC – Computer Numerical Control

GUI – Graphic User Interface

IC – Integrated Circuit

PWM – Pulse Width Modulation

SSF – Simulated Salivary Fluid

SGF – Simulated Gastric Fluid

SIF – Simulated Intestinal Fluid

PCB – Printed circuit board

NiCr – Nichrome

I^2C – Inter-Integrated Circuits

LIST OF FIGURES

LIST OF TABLES

1. MOTIVATION

The development of new drugs and food supplements is a lengthy process that requires extensive studies regarding the evaluation of their efficacy and safety. Critically, aiming for the comfort of the consumer, the oral route of administration remains, by far, the most preferred. However, for orally ingested food or drugs to exert their effects they must first resist the harsh environment found during digestion through the gastrointestinal tract (GIT) – coined bioaccessibility, and second, be absorbed at the intestine to the blood circulation so that they can reach their target organs. In addition, studying the safety of these compounds is a key step leading to their approval for human consumption.

Digestion is a complex, multistage process and its study *in vivo* is not always ethically, economically, or technically viable. Alternative *in vitro* models have been developed to simulate the human gastrointestinal tract (GIT) and assess biological responses. *In vitro* methods, starting from simple static models to more complex dynamic models, have been used as reliable tools to predict the behavior of tested compounds through the three main phases of digestion (oral, gastric and intestinal). In 2014 the INFOGEST COST Action[1] established an international consensus defining a protocol that has become the gold standard for simulating digestion *in vitro*, through a static approach. Despite being widely used, it is restricted to end-point assessment. This protocol, though robust and highly informative, lacks the capacity to include crucial kinetic aspects related with the gastric phase, such as gradual acidification, fluid and enzyme secretion and gastric emptying. In 2020, building up on the previous method, the COST Action INFOGEST published a semi-dynamic methodology that is more physiologically relevant, with particular focus on the gastric phase [2]. Other models, called dynamic models, include examples like the TIM[3] and SHIME[4] and allow for more accurate simulations of digestion. These models typically include dynamic addition of enzymes, pH control, several chambers, gastric emptying, programmable secretion times, and others. As a drawback, these models are bulky, complex and require the use of large volumes of both sample and reagents. This makes them not applicable to valuable compounds present in minute concentrations, as is the case of newly developed nanoformulations aiming to be used as drugs or food supplements. Moreover, those approaches do not include cells to evaluate the biological responses of digested samples.

Thus, there is a gap between the complex and high-consumption dynamic models and the robust, but simple static models. The aim of this project is to bridge this gap by developing an automated miniaturized system capable to simulate dynamic digestions. This will allow to reduce sample and reagent usage, operator inputs, and to downsize lab occupation. The device will include feedback controls for temperature and pH, gastric emptying, and continuous data monitoring as a platform to simulate dynamic human digestion.

2. INTRODUCTION

2.1. Current approaches for *in vitro* digestion

In vitro digestion methods have been used extensively to assess the fate of orally consumed products. These methods allow to assess possible hazardous effects of new food components, supplements or drugs while reducing the use of animal models, and human testing, at lower costs.

Although current *in vitro* methods cannot fully replace *in vivo* models, they are effective for early stage research and help minimize *in vivo* testing significantly. Many approaches have been developed to simulate the digestion process *in vitro*, and each has its own advantages and downsides. Each approach studies the compounds in the samples differently including chemical changes (hydrolysis of lipids, proteins and/or polysaccharides), location changes (release of encapsulated components, competitive adsorption processes, multilayer formation), and structural changes, such as the breakdown of specific structures, aggregation, droplet coalescence or disruption [5]. Most existing *in vitro* digestion models share a common ground on the used enzymes and other biomolecules involved in the digestive process. These include pepsin, pancreatin, trypsin, chymotrypsin, peptidase, α-amylase, lipase, bile salts and mucin, which can be of human, animal, or plant origin.

In general, digestion protocols contemplate three phases: oral, gastric, and intestinal/duodenal which may vary slightly in duration. Oral or salivary spans from 2 to 15 minutes, gastric from 30 to 180 and intestinal from 60 to 360 minutes [6]. The pH at which each phase is run is also the subject of great variability with different approaches existing with ample result spectra and objectives [7]–[10].

Recently, the standardized protocol for in-vitro digestion from the INFOGEST COST action has gained wide acceptance due to its simplicity and high reproducibility [11]. The protocol was developed based on the expertise of many scientists worldwide and standardizes the simulant fluids, temperature, pH and time for each digestion phase. It also specifies enzymes to be used and expected activity levels. Besides its reproducibility, it provides intra laboratory reliability while keeping methodology simple as shown by Egger *et al.* [12].

However, the INFOGEST protocol is still limited by its simplicity, lacking a good replication of the dynamic conditions of the digestive process, including gastric emptying and dynamic enzyme secretion rates. Nevertheless, the INFOGEST protocol has demonstrated a high reliability and constitutes a good starting point for the development of more complex methods [13],[14].

2.2. Dynamic and semi-dynamic methods

Dynamic digestion methods try to cover the demand for faithful replication of human digestion *in vitro*. When compared to static protocols like the INFOGEST protocol, these methods, are more physiologically relevant, although they are more complex and present technical challenges [15].

Several dynamic models have been established, all of them requiring big assemblies of control machinery and reaction chambers as well as relatively large amounts of samples and other consumables/reagents. Two of these models are the TIM [3] and SHIME [4] which have previously demonstrated to mimic human digestive conditions with high reliability [16], [17]. These are not the only examples and other models for dynamic digestions have also gained acceptance and are now widely used [18].

Between the fully dynamic and the static methods, semi-dynamic methods have also been proposed. These aim to replicate dynamic digestion as closely as possible while still maintaining low complexity. One relevant example of these semi-dynamic digestion methods was described by Mulet-Cabero, [2] where the

INFOGEST protocol was followed but modifications were made to include gastric emptying, the gradual addition of the sample to be processed, as well as continuous pH adjustments and addition of simulant fluids. This semi-dynamic protocol has proven to be a good middle ground between the two main in-vitro digestion methods (static and dynamic) [19]–[22].

Considering the recurring trend for miniaturization, sustainability and low-cost platforms, digestion-on-chip systems have come to the fore. Pim de Haan *et al.* [23] reported a method that produced good results with fluorescent reporter molecules and showed complete digestion of lactoferrin. However, this device was still overly simplified and although digestion occurred in continuous flow there was no gastric emptying, dynamic addition of enzymes or real-time adjustment of pH. Thus, though the device provided the advantage of lower reaction volumes it effectively behaved like a static digestion protocol.

2.3. Temperature and pH measurement techniques

Temperature and pH controls are important for in-vitro digestion simulation given that they are two of the main factors that affect test compounds and, critically, enzyme activity. In the INFOGEST protocol, pepsin and pancreatin are used to digest protein samples. For example, pepsin is only active at a low pH value within the range of pH 2.5 but maintaining high function (70%) up to pH 4.5 [24]. Previous studies evaluated the activity of pepsin at different temperatures and showed that at temperatures higher than 40 ºC inactivation was inevitable [25].

2.3.1. Temperature

There are numerous approaches for temperature monitoring: thermocouples, resistive temperature devices, infrared sensors, bimetallic devices, thermometers, and others. Integrated circuits (IC) temperature sensors capable of doing digital communication have less issues with noise when compared to more analog systems and require less sensitive and simpler data acquisition systems, given that all the readings are converted inside the IC and therefore the digital signals are simple to read [26], [27].

There are also many ways to deliver controlled heating such as resistors, heat-pumps, and magnetic fields. For example, Nichrome (NiCr) is a type of alloy that has a high melting point and opposed to other metals, is a bad conductor given its high resistivity ($100\text{-}150\text{x}10^{-8}$ Ω.m), which makes it a great material for heating. This alloy has even been used in microfluidics applications as a micro-heater sputtered onto ceramic plates[28].

In order to control temperature with tight temperature tolerances, the most widely used method for power delivery calculation is the proportional-integral-derivative (PID) control method. This method calculates how far the measured temperature is from the set point, time elapsed between readings, change in error, and using pre-determined constants for P, I, and D, determines how much power to apply to the control method used [29]. This approach is also utilized by some microfluidics systems such as the one in S. Atabakhsh *et al.* [30].

2.3.2. pH

There are two main approaches to measure pH, colorimetric analysis, usually using pH paper strips, and electrochemical methods. The most common probe for electrochemical measurements consists of a glass electrode together with a reference electrode (Figure 1) [31]. This method attains highly accurate measurements with the electric potential escalating with H^+ concentration. The probe incorporates an internal buffer (typically 3M KCl), which can be refilled through an opening. The buffer connects to a reference electrode, usually made from silver/silver chloride (Ag/AgCl), maintaining a constant reference potential at this electrode. An inner chamber has an electrode fused with a glass membrane (glass electrode), which is permeable to the solution being measured, and selectively binds H^+ and other cations thus changing its conductivity [32].

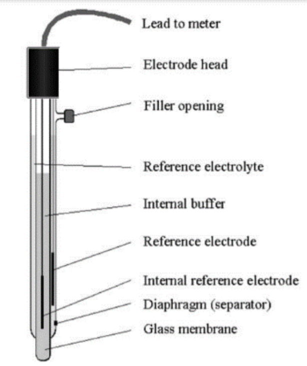


Figure 1 – Example of a typical glass pH probe

Recently, with the requirement of miniaturization, new techniques based on metal oxides, such as iridium oxide (IrOx) or ruthenium oxide (RuOx), have been developed. These metal oxide probes provide high precision and work by chemisorbing water onto the metal oxide surface enabling interaction with protons in solution (hydronium) changing their surface potential, which is then correlated with pH (Figure 2). Metal oxides have extremely fast pH responses, which is useful for continuous pH measuring. They can also be coupled with the same Ag/AgCl and 3M KCl reference solution, further increasing stability and precision and thus being a good solution for short-term measurements. [33]

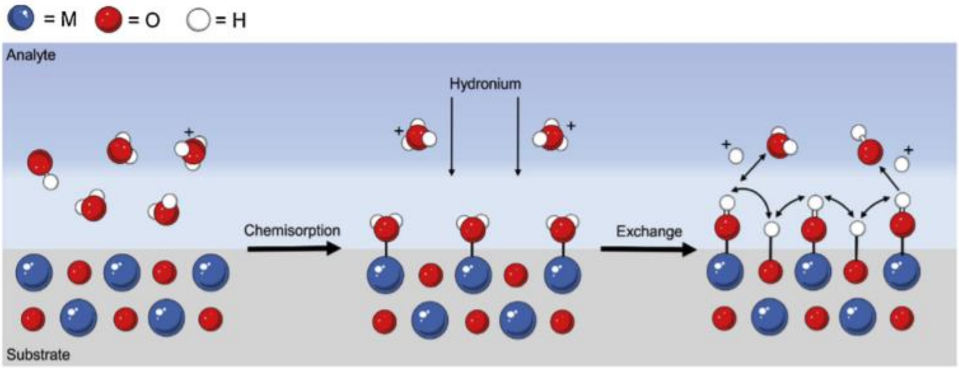


Figure 2 – Mechanism for pH measurement based on metal-oxides (adapted from [33])

2.4. Objectives of the project

The overarching goal of the project is to develop a modular microdevice to mimic the complete gastrointestinal tract that would enable to evaluate the digestibility and intestinal permeability of bioactive compounds. This work aimed to develop the *in vitro* digestion module, with the following specific objectives:

- To develop and fabricate a miniaturized *in vitro* digestion platform that will allow automated simulation of human digestion with reduced sample and reagent volumes.
- The platform should include real-time pH and temperature monitoring with integrated feedback loops for heating and pH adjustments.
- The model will work in a semi-dynamic mode including gastric emptying from the gastric to the intestinal chamber.

To achieve this, devices were designed using 2D drawing software aiming for fabrication following a layer-by-layer assembly using double-sided tape and polymethylmethacrylate (PMMA) sheets machined by CNC micromilling or laser cut. PMMA is fairly inert and has been used other digestion on chip studies as well as in devices working with cells [34]–[36]. Thus, it represents a good alternative to the ubiquitous polydimethylsiloxane (PDMS), which is typically used in lab-on-chip applications but has been shown to absorb small molecules potentially leading to altered results [37]. This has been also demonstrated in early work developed within the research group.

Device planning included the seamless integration of commercial temperature sensors and in-house built pH sensors to monitor two key parameters governing enzyme activity. The sensors can be connected to custom-built electronics for data acquisition and an Arduino circuit board. The Arduino can be further used to i) drive heating elements to maintain the temperature in the reaction chambers stable through the digestion experiment, and ii) to control syringe pumps used to add acid of alkaline solutions to adjust the pH at the different stages of digestion.

To validate the device, the use of a report molecule such as the quenched, fluorescently labelled casein derivative used by de Haan and colleagues [23] provides a means to compare directly the extent of on-chip digestion with current static *in vitro* protocols. As a reference *in vitro* protocol, the standardized method established within the INFOGEST cost action consortium [1], [11] represents the current gold standard static method and was the method of choice to validate the device developed within this work.

3. MATERIALS AND METHODS

3.1. **Device Fabrication**

Devices were fabricated by assembling different machined layers of polymethyl methacrylate (PMMA). The layers were fabricated using a combination of laser cutting (Widlaser LS1390 Plus with LaserWorkV6 software) and CNC micro-milling (Flexicam Viper 606 with ArtCam software) and bonded using pressure-sensitive double-sided tape (467NM 200MP; 3M, Saint Paul, MN, US). A Polydimethylsiloxane (PDMS) lid was used to avoid evaporation in the reaction chambers. The lids were fabricated by pouring PDMS (SYLGARD™ 184 Silicone Elastomer; Dow Chemical Company, Midland, Michigan, EUA) at a 10:1 ratio onto a PMMA mold previously fabricated by CNC micro-milling. The PDMS was degassed and cured in an oven at 65ºC. Openings were punched through the PDMS using Biopsy punches (Kai-Europe GmbH, Germany) so that tubing and sensors could be inserted in the digestion chambers.

3.2. **pH Monitoring and Control**

Iridium Oxide was deposited onto a Titanium wire and silver chloride was deposited onto a silver wire by electrodeposition. These Ag/AgCl electrodes where then encapsulated in a 10uL pipette tip containing a 3M KCl reference solution. (see Appendix D). A custom-built electronic circuit was used to interface two pairs of pH sensors by reading the differential electric potential.

An Arduino microcontroller was used to control two custom-made Syringe pumps (adapted from https://github.com/DropletKitchen/pumpsn17). Syringes were placed on each pump and connected with tubing to the chambers in the device. The syringes contained 1M HCl and 1M NaOH for pH adjustment in the gastric and intestinal chambers respectively.

3.3. **Temperature Control**

Small inter-integrated circuits (I^2C) digital temperature sensors (MAX30205) were used to measure the temperature in the reaction chambers during digestion. These sensors were soldered onto a printed circuit board (PCB) design using Autodesk's Eagle software and manufactured by Eurocircuits GmbH. The sensors were controlled using the same Arduino microcontroller used to control the in-house built syringe pumps. The temperature measurements were then used in a feedback loop to drive electrically insulated Nichrome (NiCr) wires placed at the bottom of the chambers, here used as heating elements due to their high resistivity, and keep the temperature stable at 37 °C.

3.4. **Mixing**

Constant mixing was achieved using small magnetic stirrers placed in the chambers driven by two computer fans controlled by pulse width modulation (PWM) with diametrically polarized magnets attached on top. It was important to maintain a relatively slow mixing speed through PWM to avoid enzyme degradation.

3.5. **Code and user interface**

Programming was done both in Arduino and MATLAB (MA, US) and the communication between the software and the devices (Microcontroller and Data Acquisition box respectively) secured by USB connection. A Graphical User Interface (GUI) was also built in MATLAB for displaying pH sensor data along with other information in the console such as time elapsed, current pH and approximate amount of HCl/NaOH added for pH adjustment. In Arduino, the code was used to drive the syringe pumps and control the temperature inside the digestion chambers.

3.6. **Experimental Procedure**

The standardized INFOGEST digestion protocol was followed as a control reference to all experiments. This protocol (see appendix A) uses three different simulant fluids that mimic the human salivary, gastric, and intestinal digestive fluids. These fluids were prepared in advance according to the recipes in the same appendix. To validate the digestion device, a commercial casein derivative labelled with BODIPY® TR-X dye was used. The dye is quenched by proximity becoming fluorescent upon hydrolysis (λex=589 nm, λem=617 nm), thus working as a digestion reporter molecule to determine protease activity, being proportional to the digestion rate and measured using a BioTeK® Synergy H1 microplate reader (Winnoski, VT, USA) and Gen5.1 software. Digestion in the oral phase is a 2-minute process and does not lead to casein digestion. During the gastric and intestinal phases, 100-µL samples were collected every 15 minutes to evaluate digestion kinetics.

The pH probes were calibrated prior to the start of each experiment and inserted along with the tubing connected to the HCl and NaOH syringes through holes in the PDMS lid. After reagents are added to the chambers, the system automatically adjusts pH and temperature. While running, the software adjusts pH to 3 (using HCl) during gastric digestion and heats the chamber to 37 °C. After 120 minutes (duration of the gastric digestion), digestive fluids are transferred from one chamber to the other and pH is then readjusted to 7 for the intestinal digestion, which also has a duration of 120 min.

3.7. **Peristaltic pump**

A simple peristaltic pump was fabricated using PDMS and PMMA following the same fabrication techniques as those used for the main device. It consisted of a 5 mm machined PMMA layer with a rectangular groove for a PDMS piece, and a long channel with three chambers. PDMS was used to create a thin membrane which is bonded onto a rectangular piece of PDMS that fits into the PMMA groove, with three chambers that can be pressurized using a Fluigents MFCS™-EZ Pressure controller to deform the membrane that displaces the liquid and thus allows peristaltic motion.

4. RESULTS AND DISCUSSION

4.1. Device design, Fabrication, and operation

Although PDMS is the most widely used polymer for the fabrication of microfluidics devices, it has been shown to significantly absorb small molecules and thus affect the experimental outcome of analytical assays. To avoid this, polymethylmethacrylate (PMMA) was used for fabrication. In addition, PMMA is advantageous for being easily machined using rapid prototyping techniques such as laser cutting or CNC micro milling.

The original device design for the digestion devices is shown in Figure 3. It consisted of two chambers connected by a valve and side channels to pump in the different fluids needed for the digestion process.

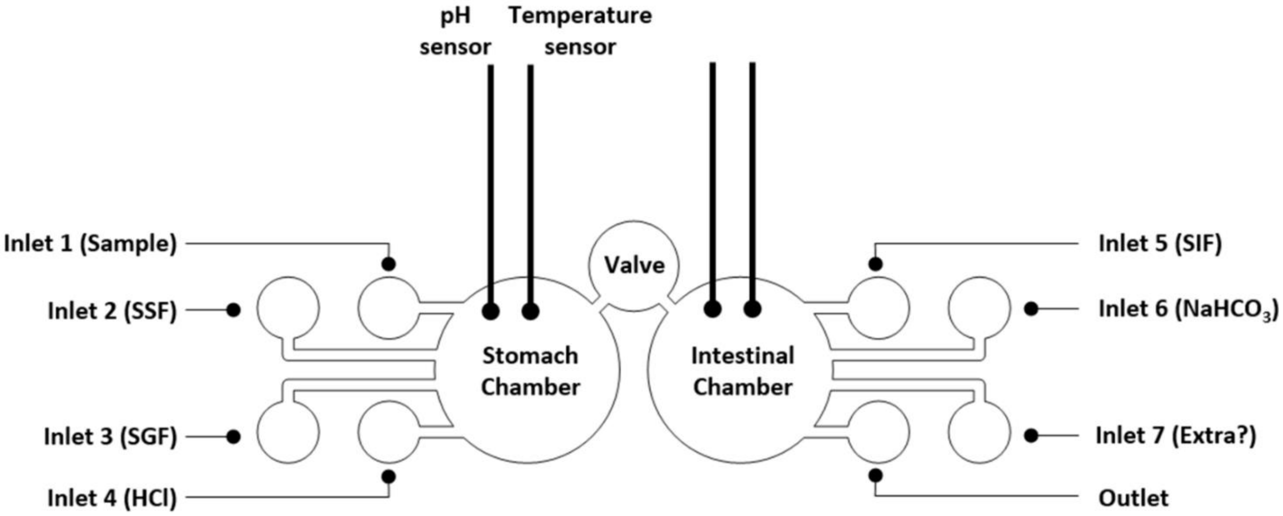


Figure 3 – Original design with two main reaction chambers “Stomach chamber” and “Intestinal Chamber” along with a valve that would control gastric emptying, and several smaller chambers connected to the former via channels that act as inlets and outlets for samples, simulant fluids, acid, base and some extra if needed. It is also possible to see two sensors in each chamber for pH and temperature measurements and control.

The devices were designed so that the reaction chambers would fit in the positions where the standard 24-well plates sit inside a Microtiter Plate Reader and could therefore be directly inserted in them. Device and chambers had the same dimensions of a standard 24 well-plate. The connections were threaded so that tubing could be inserted leading to syringe pumps. A micro-valve was used to allow the liquid to flow from one chamber to the other by gravity in a controlled manner, in order to mimic the gastric emptying.

However, the design was modified to simplify its operation and improve the technique to drive the gastric emptying. The final design is shown in Figure 4 and Figure 5 shows all the components of the complete setup.

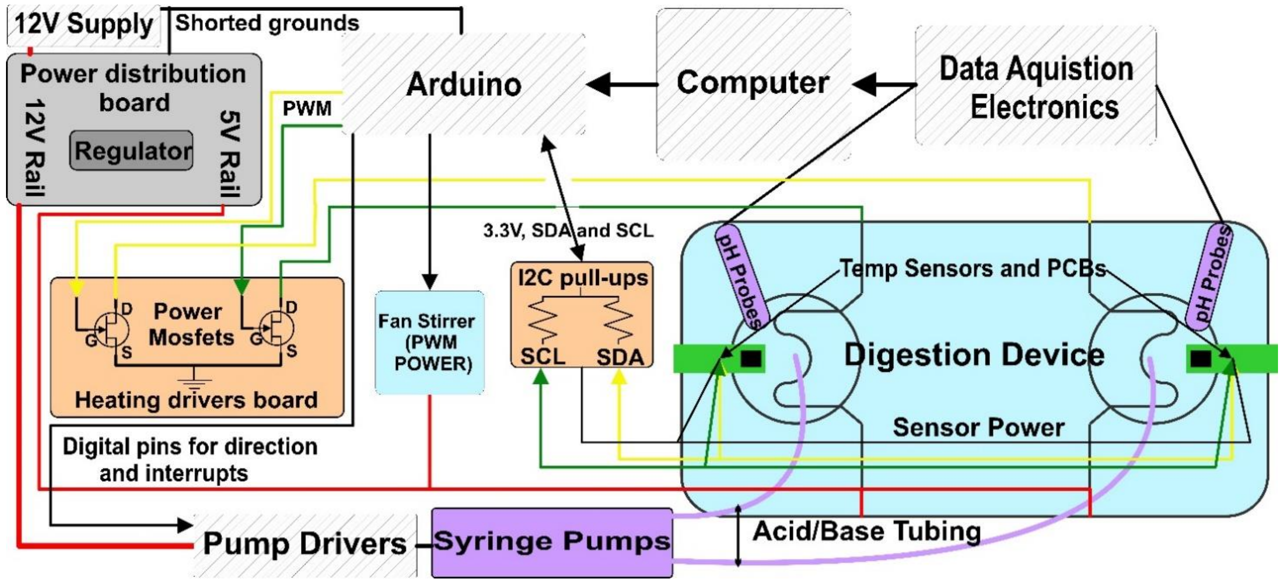


Figure 4 – Electrical Schematic of the hardware used (without peristaltic pump) for power regulation (grey), temperature control (orange), pH control (purple), and fan stirrer (cyan)

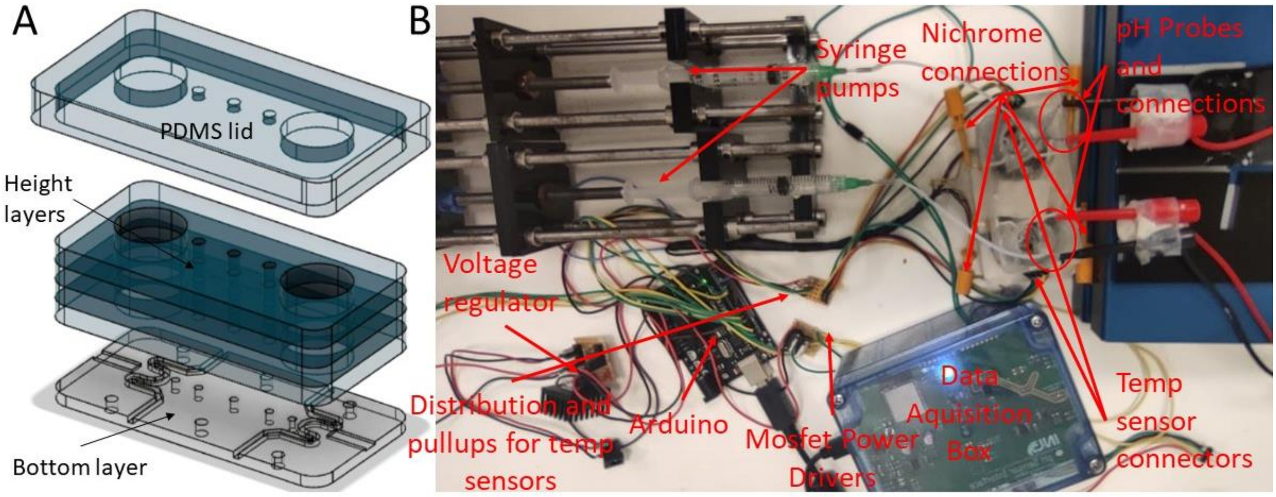


Figure 5 – A) 3D model of the final design achieved and B) picture of the experimental setup along with labels.

In the final design, all features for real-time control of pH temperature and a microfluidic peristaltic pump are placed at the bottom figure 5 and 6. The device has a dimension of 77x39mm in area and 21mm height, which becomes 26mm with the peristaltic pump attached to the bottom. Chambers have a diameter of 17mm and a height of 16mm and can fit a maximum volume of 3mL.

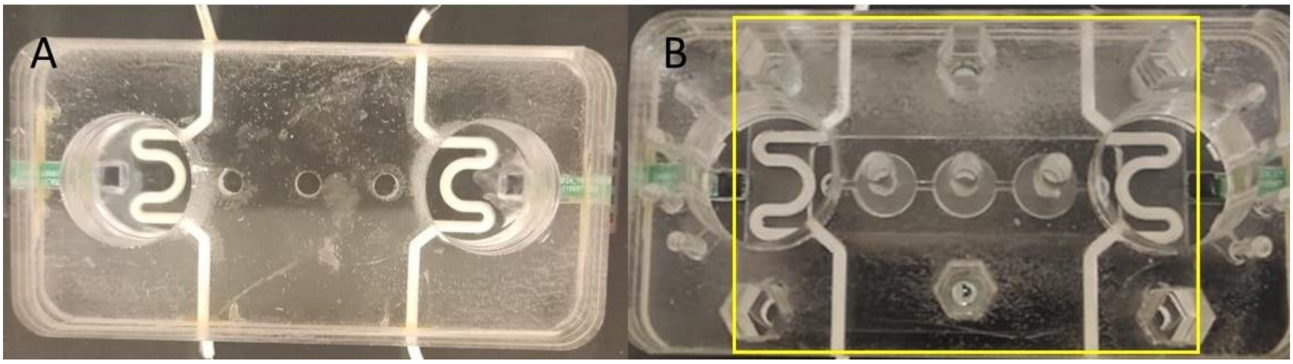


Figure 6 – Top-view of digestion device (A) without and B) with peristaltic pump integrated (highlighted))

4.1.1. Bottom Plate

The bottom layer is a critical part of the design since it is where the temperature control is integrated, where the magnetic stirrer sits and, where liquid can flow to the peristaltic pump below when gastric emptying is used. Improvements were made to the initial design to optimally integrate the heating elements and temperature sensors. Originally glass microscope slides with deposited resistors (Figure 7) placed at the bottom of the chambers were used for heating. However, the deposition technique together with the required electrical insulation of the heating elements proved complicated and would require extensive cleanroom processing.

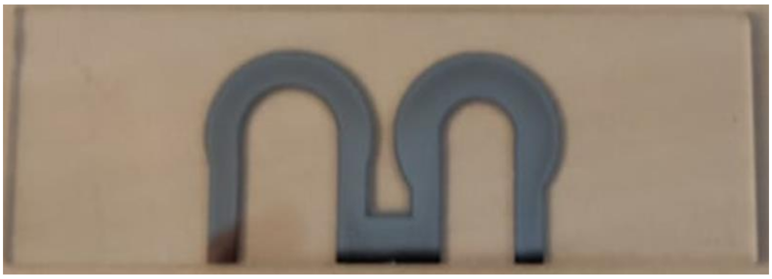


Figure 7 - Sputtered Resistors on glass microscope slide

Instead, NiCr wires insulated with silicone from 24 AWG wire were chosen as heating elements. CNC micro milling enabled to fabricate bottom plate grooves where the wires could be incorporated. Grooves were also made for the PCBs with soldered temperature sensors. The grooves for the NiCr wires were 1.5-mm deep and 1.5-mm wide and the grooves for the PCBs were 13-mm wide and 0.7-mm deep. This plate can also have holes drilled for the connection to the peristaltic pump (figure 8).

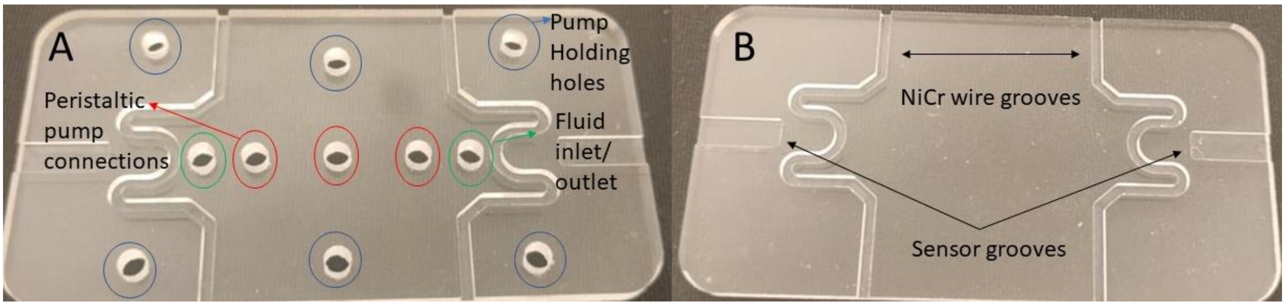


Figure 8 – Bottom layer of the digestion device (A) with and B) without the possibility of attaching the peristaltic pump)

4.1.2. Middle Plate/Height Layers

The final design for the middle plates consisted mainly of two circles that constitute the reaction chambers. The height was controlled depending on the thickness and number of the PMMA layers used. The layers were typically fabricated from 5mm acrylic and if gastric emptying was to be used, extra features were added including three small holes for connecting the pressure controller to the peristaltic pump and, six hexagonal through-holes to fit nuts that hold the peristaltic pump in place. Devices were typically made off 4 layers providing a maximum liquid capacity of 3mL to each chamber. The three smaller holes are present on every devices top layer as they are also useful to hold the PDMS lid in place.

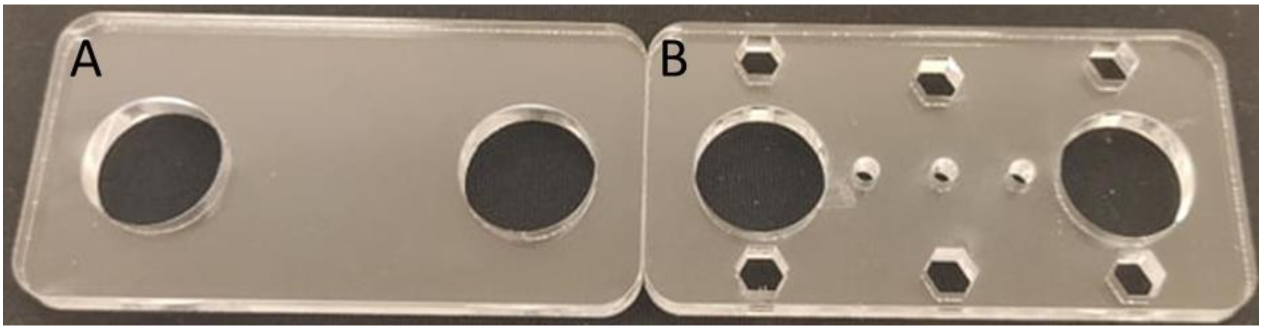


Figure 9 - Height layer of the digestion device (A)without and B) with peristaltic pump attachments)

4.1.3. Lid

Preliminary experiments showed that liquid evaporation inside the chambers was significant enough to impact fluorescence readings. To solve this issue, a PDMS lid was fabricated to seal the chambers during the digestion experiments. Several designs were tested, and the final design consisted of a 2-mm thick lid with 5-mm overhangs on the outside of the device (to ensure that no condensation could escape) together with 4-mm inserts that fit inside the digestion chambers for the same reason. There is also a 4-mm diameter hole on top of each digestion chamber so that samples and reagents could be added or removed. These were sealed with PDMS or PMMA plugs during the experiments. A biopsy punch was also used to open two additional holes on each of the chambers. A bigger one, used for the pH probes, and a smaller one for inserting the tubes connected to the syringes with HCl and NaOH solutions.

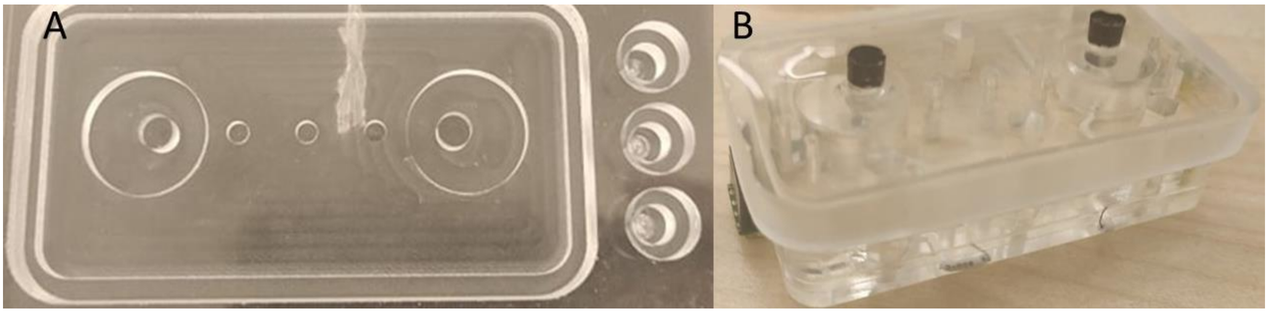


Figure 10 – A) Lid mold and B) assembly in device with PMMA plugs

4.1.4. Peristaltic Pump

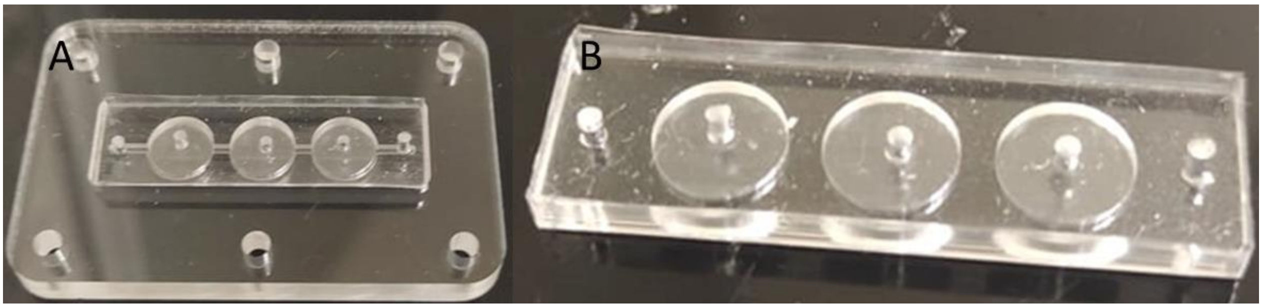


Figure 11 - A) Full peristaltic pump assembly (PDMS + PMMA) B) close-up view of actual PDMS peristaltic pump

A pressure controller was used to compress a PDMS membrane that was coupled to PMMA chambers connected by a horizontal channel. By actuating them sequentially (Table 1), liquid could be pushed and moved as happens in the better known motorized, circular peristaltic pumps. The PDMS membrane was bonded to a bigger, rectangular piece of PDMS (Figure 11 B), which was used to simplify handling and to provide support to the connectors used as inputs for pressurized air. The pump was characterized by moving an amount of fluid similar to the one used in a regular digestion experiment from chamber 1 to chamber 2. It was shown that it could transfer about 1 mL in every 10 minutes, (Figure 12A) which corresponds roughly to 50% of the contents of the chamber. The fluid transfer rate was linear indicating that fluid transfer is controllable. Figure 12B shows the experimental setup for the characterization.

Table 1 - Diagram for pressure controller functioning to power peristaltic pump. In black, pressurized chamber and in white, unpressurized chambers. The cycle repeats itself for as long as the pump is on.

	Chamber 1	Chamber 2	Chamber 3
Step 1	○	●	●
Step 2	○	○	●
Step 3	●	○	●
Step 4	●	○	○
Step 5	●	●	○
Step 6	●	●	●

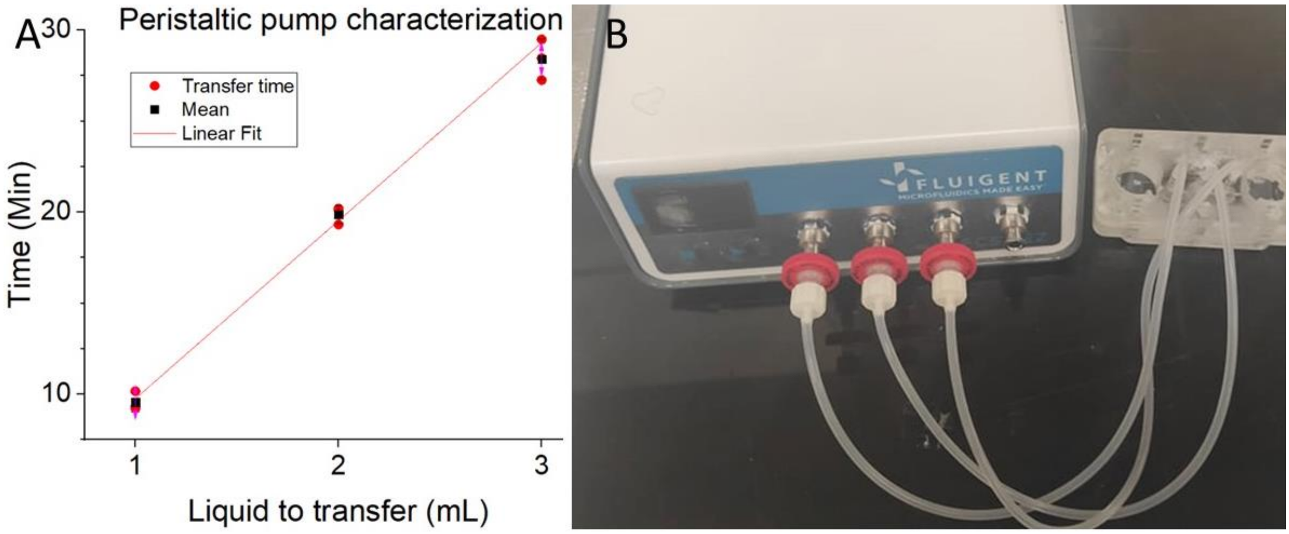


Figure 12 - A) Peristaltic pump characterization (using n = 3 and 3 different volumes (1,2,3)) linear fit (y=mx + b) was traced yielding (r2=0.996, m = 9.77 and, b = -0.01). Experiments were performed by placing a set amount of liquid in chamber one, starting a timer and observing how long liquid took to be pumped to the second chamber. B) Experimental setup for pump characterization showing the pressure controller and the digestion device along with the three pieces of tubing and connectors.

4.1.5. Magnetic Stirrer

A magnetic stirrer was custom-built to be placed under the chambers and drive stirring magnets. Also here, a layer-by-layer approach was followed with each layer laser cut and bonded using double sided tape. The layers could fit controllable fan motors and wiring. The motors were controlled by PWM and powered with 5V. The PWM signal was input via Arduino digital pins and kept at the lowest possible settings to achieve slow but constant mixing so as not to lead to any possible enzyme degradation.

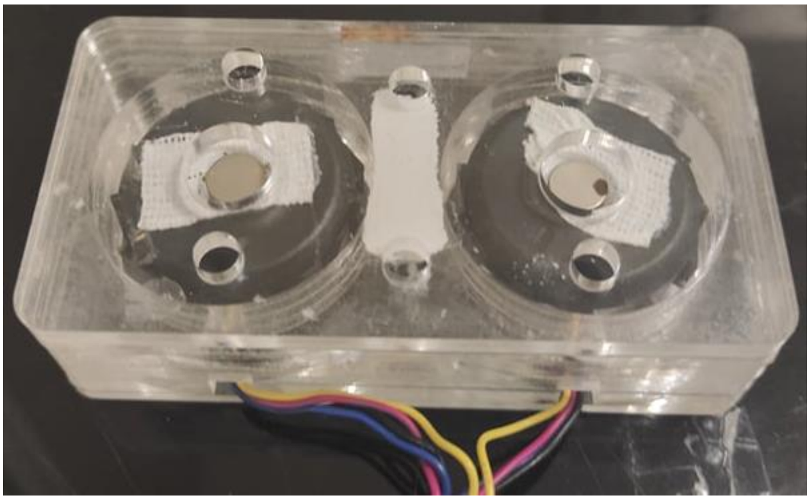


Figure 13 - Fan powered magnetic stirrer consisting of two 5V computer fans with the blades removed, two pieces of double-sided tape holding diametrically polarized magnets and a PMMA frame around it with holes for the fan wires with a top plate containing holes for the nuts that hold the peristaltic pump to insert in and keep the device from moving.

4.2. **Control Software and Programming**

The full scripts can be found in Appendix D – G.

4.2.1. MATLAB

MATLAB was the software mainly used for data acquisition. Data was collected from the pH probes at a 3 Hz sampling rate. Simultaneously, data from both temperature sensors was acquired so that pH and temperature readings could both be displayed in real-time. The PID algorithm was run along with sensing so that the solutions could be heated, and the pH adjusted automatically and according to the readings. Communication between Arduino and MATLAB was made via a string. It had a starter and finisher ('<', '>') and the information for Arduino is separated by colons ':'. For the final code, the string consisted of a letter and four separate digits indicating what Arduino should do in the following order (A – Gastric chamber heating, B – Intestinal chamber heating, C – no heating), two zeros that used to indicate duty cycle for each chamber but that were not used in the final design given that sensor reading was done directly by the Arduino, and two more values indicating the number of steps the two syringes should take to maintain the pH set to each chamber. This is an example of a string that MATLAB would send running GASTRIC digestion with the pH being adjusted: <A:0:0:1000:0>.

4.2.1.1. pH Control Algorithm

pH probes were calibrated prior to the experiments given that different probes have different mV/pH ratios and y-intercept values. For this, three standard pH buffer solutions from Mettler Toledo were used (4.01, 7.00 and 9.18). After calibration, the voltage readings corresponding to pH 3 and 7, which are the values required for gastric and intestinal digestion respectively, were calculated in MATLAB.

The algorithm used for pH control detected how far from the set points the current pH readings were and calculated how many steps from the stepper motor driving the syringe pumps would be needed to reach them. It is important to note that depending on the phase of digestion, a different algorithm was used to adjust pH as for with the addition of acid, pH goes down while when base is added, pH goes up. The algorithm was designed so that more steps were driven when the current value was further from the set point so that initial adjustments were quicker. Stepper motor steps were converted into volume following a simple calculation. The NEMA17 stepper motor used to fabricate the syringe pumps rotates 0.9º per step. The motors were actuated with 1/16 micro stepping to improve precision. Thus, each step represents a rotation of $0.9° * \frac{1}{16} = 0.05625°$. The motor was attached to an 80-threads per inch precision screw resulting in a displacement of 317.5 µm per revolution. Thus, the displacement per step can be calculated as $\frac{317.5*0.05625}{360} = 0.0496\ \mu m.$ Plastic 1-mL syringes from BD with an inner diameter of 4.78 mm and thus an area of 17.95 mm^2 was used. A displacement volume per step can be calculated using the area of the circle and the piston advance per step which is $0.0000496 * 17.95 = 0.00089\, mm^3$ or 0.89 pL. The adjustments were done in 40 measurement intervals so that the pH had time to settle before the next number of steps was calculated.

4.2.2. Arduino

4.2.2.1. Temperature control

The Arduino was initially used to receive instructions from MATLAB and drive the syringe-pumps and heating elements. However, due to problems with data-acquisition, it was necessary to also use Arduino to monitor the temperature. This was done because the ESP32 microcontroller from the data acquisition box appears to be too sensitive in its I^{2}C data lines as opposed to the Arduino ATmega328P. To read data from the sensors two Arduino libraries were used. These were Wire, a library used for I^{2}C communication, and Protocentral_MAX30205, which can send messages to the sensor and receive responses by calling the sensor and using the getTemperature function. A modification was done to this library so that Arduino would look for two sensors with different I^{2}C addresses and they could be called and addressed independently. Depending on the character present in the string received from MATLAB, a different sensor is called, and thus a different chamber is heated using the respective PWM digital pin. This programme also had the capacity to not heat any of the two chambers and to display the temperature of both chambers if Arduino IDE was open. When a string is received in Arduino, the temperature is read and the PID calculated to achieve the set-point, 37ºC. PID control was chosen to achieve a constant temperature with tight tolerance. Liquid contents varied as samples were taken/reagents added so a constant power delivery wouldn't achieve the desired constant temperature. The PID function uses Millis (a function available in Arduino that keeps track of time in milliseconds) to keep track of time between readings. It calculates set-point deviation, the error change and, using this information and the predetermined P, I and D constants, calculates the duty cycle that should be used for each chamber digital pin. The duty cycle can vary from 0 to 255 but in this case the values were constrained to 0 to 75 as high localized heating could be problematic and cause enzyme denaturation. I values were also constrained because very high I error accumulated causing stabilization issues.

4.2.2.2. pH control

The Arduino was also used to drive the syringe pumps that added HCl or NaOH to adjust pH during digestion experiments. For this, the AccelStepper library was used. Digital pins were designated for each pump and a MaxSpeed, Speed and Acceleration were defined. When the step values were received, a "power" function which ran a move command Move("steps") indicating number of steps followed by RunToPosition was used. After several cycles the pH level was achieved and maintained throughout the digestion.

4.3. Heating elements, power supply and signal connections

In a first approach, heating was achieved by a 500 mW, 100Ω resistor coated with nail polish and driven by a MOSFET. This approach proved hard to integrate in the devices and, in addition, components from the nail polish would contaminate the digestion samples and/or be cytotoxic. A second approach relied on depositing resistors on glass slides. However, this brought issues as a solution had to be found for insulating the deposited resistors. Finally, NiCr wires were chosen as heating elements and insulated with 24AWG silicone wire, which is waterproof and offers heat resistance.

The whole system was powered at 12V since this was already required for running the stepper motors and their drivers. The heating elements could also be run at this power, but this would mean very fast and highly localized heating close to the wires which could result in enzyme degradation. Given that the magnetic stirrers were also run at 5V and that the Arduino could not provide enough power to heat the chambers to 37 °C, a linear voltage regulator (L78S05CV, Digi-Key Electronics, MN, US) was used to convert the 12V supply into 5V along with a heatsink to prevent overheating and consequent shut down. Two rails were soldered on a protoboard with one connected to the 12V and the other to the 5V output. The ground for both voltages was tied together with the Arduino so that there would be no fluctuations in power.

To supply the high current needed for the heating elements, two MOSFETS (STP16NF06L N-channel MOSFET, Digi-key) were soldered on another protoboard that received the 5V and PWM signal for each digital output and sent the high current PWM to the NiCr wires. These were color coded for ease of use along with the connections to the chambers and other protoboards. Another distribution board was used to pull up the signal wires connected to the chambers via 10 kΩ resistors and to connect the two signal wires coming from each chamber to the Arduino.

4.4. **PCB design and insulation**

The temperature sensors used an I^{2}C communication protocol that enabled controlling both sensors using 2 wires only. Sensor integration in the devices consisted initially of soldering the MAX30205 temperature sensors directly to wires and insulating them using an epoxy resin. However, this proved not to yield enough mechanical resistance for the handling that the devices were subjected to. A PCB was designed to avoid these issues. It both held the sensor in place and worked as a breakout board so that a simple connector could be used. The PCB connected VDD and GND for power supply and SDA and SCL for I^{2}C communication. In a first stage, the PCBs had all address buses connected to the ground, so a single address was obtained and both sensors could be addressed using a multiplexer. Later, due to noise issues arising from the use of a multiplexer, a new version was designed to use I^{2}C to address the sensors. In the new design, half of the PCBs had one of the address buses connected to VDD instead of GND so that addressability was possible. Thus, two addresses were obtained, one for each digestion chamber.

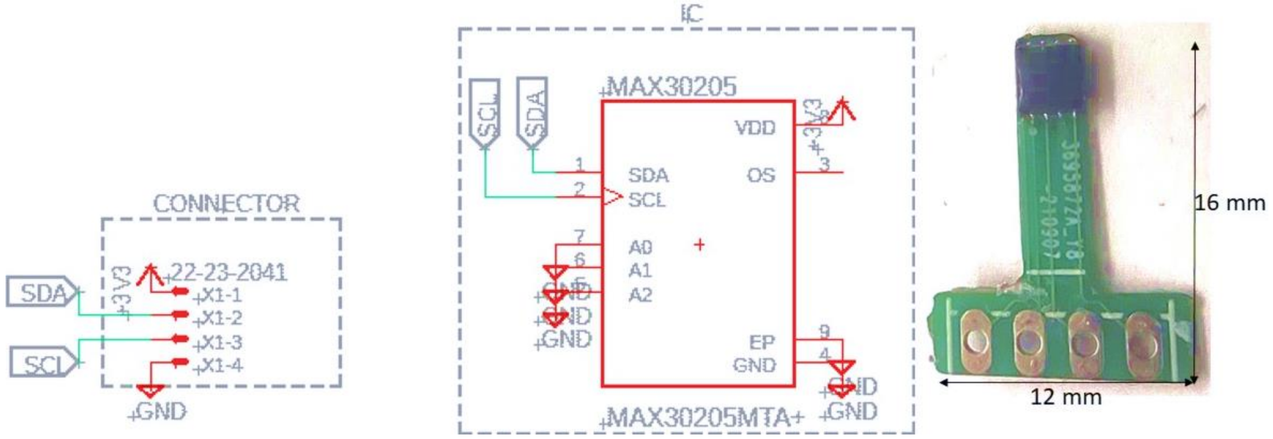


Figure 14 - PCB schematic showing the PCB with the address pins all shorted to ground (A0, A1, A2) and PCB with the MAX30205 sensor soldered on and dimensions shown alongside it.

Insulation was challenging because of the small dimensions of the PCB and sensors and because electrical signals used for I^{2}C are very sensitive. Electrical insulation also had to be achieved without affecting the thermal mass/insulation of the sensor. Many insulation methods were tested including a silicone conformal coating (RS Pro 494-714), melted paraffin, and a cellulose waterproofing formulation but none offered complete or long-lasting insulation against the harsh conditions (immersion, low pH, constant mixing) of the reaction chambers. Finally, a PDMS coating formed a strong waterproofing layer around the sensor that was not damaged during digestion experiments.

4.5. **Noise problems**

During temperature control testing, it was noticed that the sensors showed a significant noise to signal ratio, which was not observed while testing on Arduino. The noise was detrimental for temperature control as it interfered with PID causing big temperature oscillations. At first, the noise was attributed to poor sensor insulation and efforts were directed to try to improve insulation. However, the issues were still present following insulation improvements as can be seen in Figure 15.

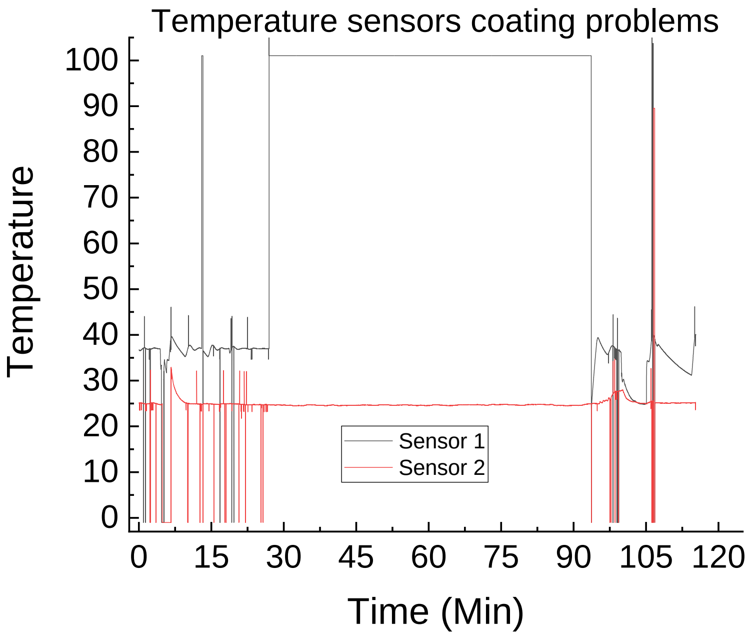


Figure 15 - Sensor noise issues during temperature control experiment. Sensor 1 is set to 37º C and sensor 2 is not heating and not immersed. Sensor 1 initially reaches the set point but does not maintain the temperature stable due to noise causing temperature to oscillate around the set point. At 30 minutes, the sensor crashes and after reseting by reconnecting (at 95 minutes), it shows ambient temperature again, rapidly increasing to the setpoint in no more than 5 minutes. Although sensor 2 was not immersed, it showed noise whenever sensor 1 did. While sensor 1 was crashed, sensor 2 did not show any noise issues, thus indicating a insulation problem.

Given that the two sensors showed noise at the same time, the source of the noise was attributed to the multiplexer used for connecting two I^{2}C devices on the same address. However, when the new PCBs with an extra address were tested (and the multiplexer removed) the problem was still present. Finally, when temperature control was performed solely from the Arduino, without using MATLAB and the data acquisition software, the noise issues were eliminated. This solution suppressed the possibility to have temperature data displayed in MATLAB.

4.6. **Control tests**

4.6.1. Temperature test

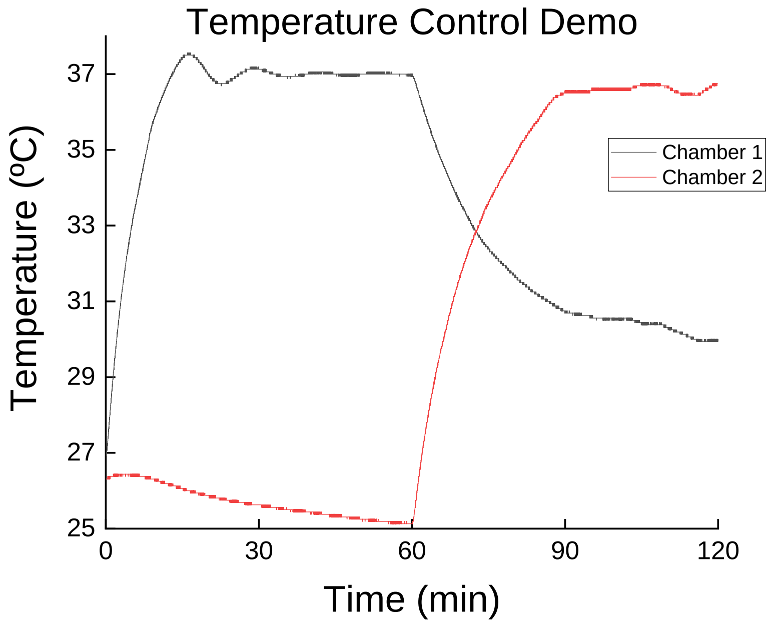


Figure 16 – Temperature control example. A program was written in Arduino using the same PID algorithm as the main code. The Arduino was set to output a temperature reading for the two chambers and during the first 60 minutes it was set to heat chamber 1 to 37º C, and during the next 60 minutes, the same was true for chamber 2.

Figure 16 shows an example of a temperature control test. The test was run over a two-hour period and each chamber was heated for 60 minutes. The chambers were filled with distilled water which is the main constituent of the digestion simulant fluids. The first chamber took around 15 minutes to reach the target temperature of 37ºC while the second took a bit longer than 20 minutes. These long times are caused by power delivery being limited via software to a value of 30%. However, since the second chamber took longer to reach the set point, it indicates that the voltage regulator is overburdened and should be better cooled. This can be avoided by using bigger heatsinks to allow the voltage regulator to maintain a low enough temperature and avoid thermal shutdown. Additionally, liquid in the second chamber will be added incrementally when gastric emptying is implemented and therefore the set-point will be reached faster. After reaching the set-points the table was kept stable within +/- 0.5 °C.

4.6.2. pH test

One of the main objectives of this project was to create a miniaturized platform for more dynamic digestion experiments. For this, controlling the pH over time is critical, particularly in the Intestinal phase where the output of the gastric digestion is added sequentially as part of the gastric emptying process. To demonstrate automatic pH control in the device, a test was run where the code was asked to adjust the pH from 7 to 3 and vice versa several times. The results are shown in Figure 17A and demonstrate the ability to successfully control the pH inside the digestion chambers within the required physiological range by precise and automated addition of 1M HCl and 1M NaOH.

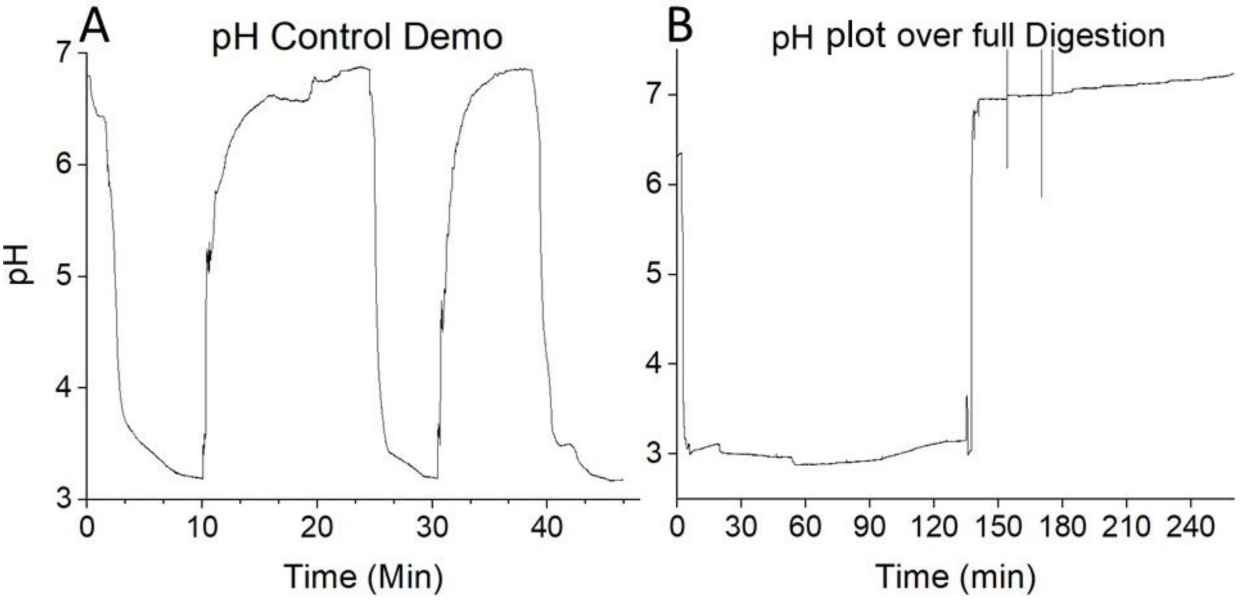


Figure 17 – A) Experiment where the device chamber 1 was filled with Simulated salivary fluid at pH 7 and, using a MATLAB program that would change the pH setpoint every time a button was pressed, setting pH to the next value (either 3 or 7). When the program did not add acid during more than one iteration, the button was pressed setting the next pH value. The pH was adjusted 5 times. B) pH plot of a full digestion experiment that failed but shows very high pH stability

Figure 17 B shows data from the pH monitoring during a typical digestion simulation. The graph shows an initial lowering from pH 6.3 to 3.0 due to the addition of HCl to the gastric chamber and later from pH 3 to 7 due to the addition of NaOH to the intestinal chamber. Some misreading's could be observed during sampling and this was due to the sensors being touched or the liquid level running too low, which could be problematic when running experiments with gastric emptying as they will require constant pH adjustment. Figure 18 shows a typical probe calibration process. The spikes observed in this figure are due to the calibration solutions being changed and the probes being momentarily unsubmerged.

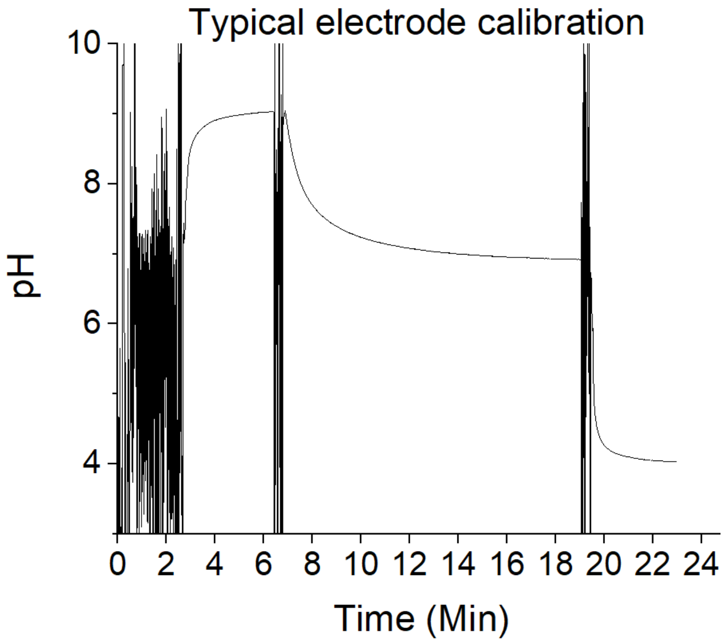


Figure 18 - Typical electrode calibration performed before a digestion experiment using buffer solutions (pH 9.21, 7.00 and 4.01) from Mettler Toledo. After the mV readings settled, the reading would be inserted in the MATLAB start function and saved. When the three points were collected, the function was restarted, and MATLAB showed the mV/pH conversion for the calibrated electrode.

4.7. Gastric digestion

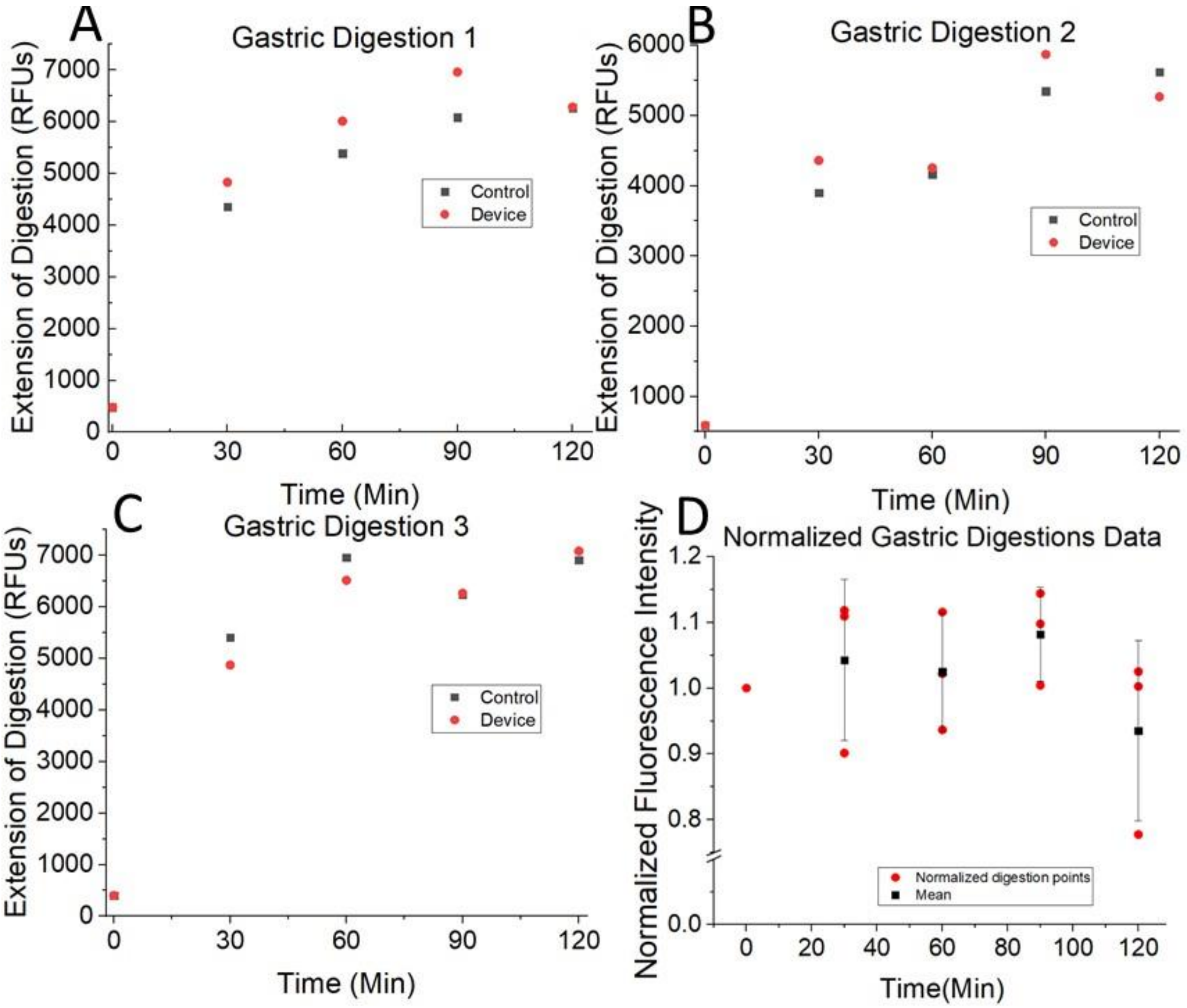


Figure 19 – A, B and C show plots of three gastric digestions with samples measured at 30-minute intervals over a period of 120 minutes. D is the plot for the normalized data calculated by dividing the results of each digestion by control digestions run simultaneously. The Figure shows Mean +/- STD.

Figure 19 shows plots of three standalone gastric digestions. The plots were then normalized by dividing the results from each experiment by control digestions following the standardised INFOGEST protocol. Figure 19D shows that the mean normalised fluorescence intensity was close to 1.0 throughout the duration of the the experiments, which indicates that there was a close aproximation of the digestions in the devices when compared to the control. Figure 20 shows the pH plot of one gastric experiment. It shows that there was significant drift in the pH readings which can be minimized by using new pH electrodes. Finally, when samples were colected (every 30 minutes in this case) there were notable spikes in the pH readings due to sensor manipulation.

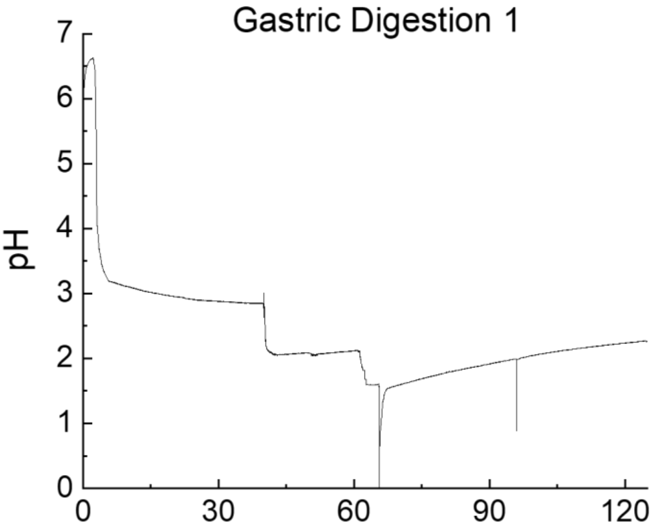


Figure 20 - Example pH plot of a gastric digestion. Significant spikes can be seen at 30-minute intervals, which coincide with sensor manipulation during sample retrieval.

4.8. **Intestinal digestion**

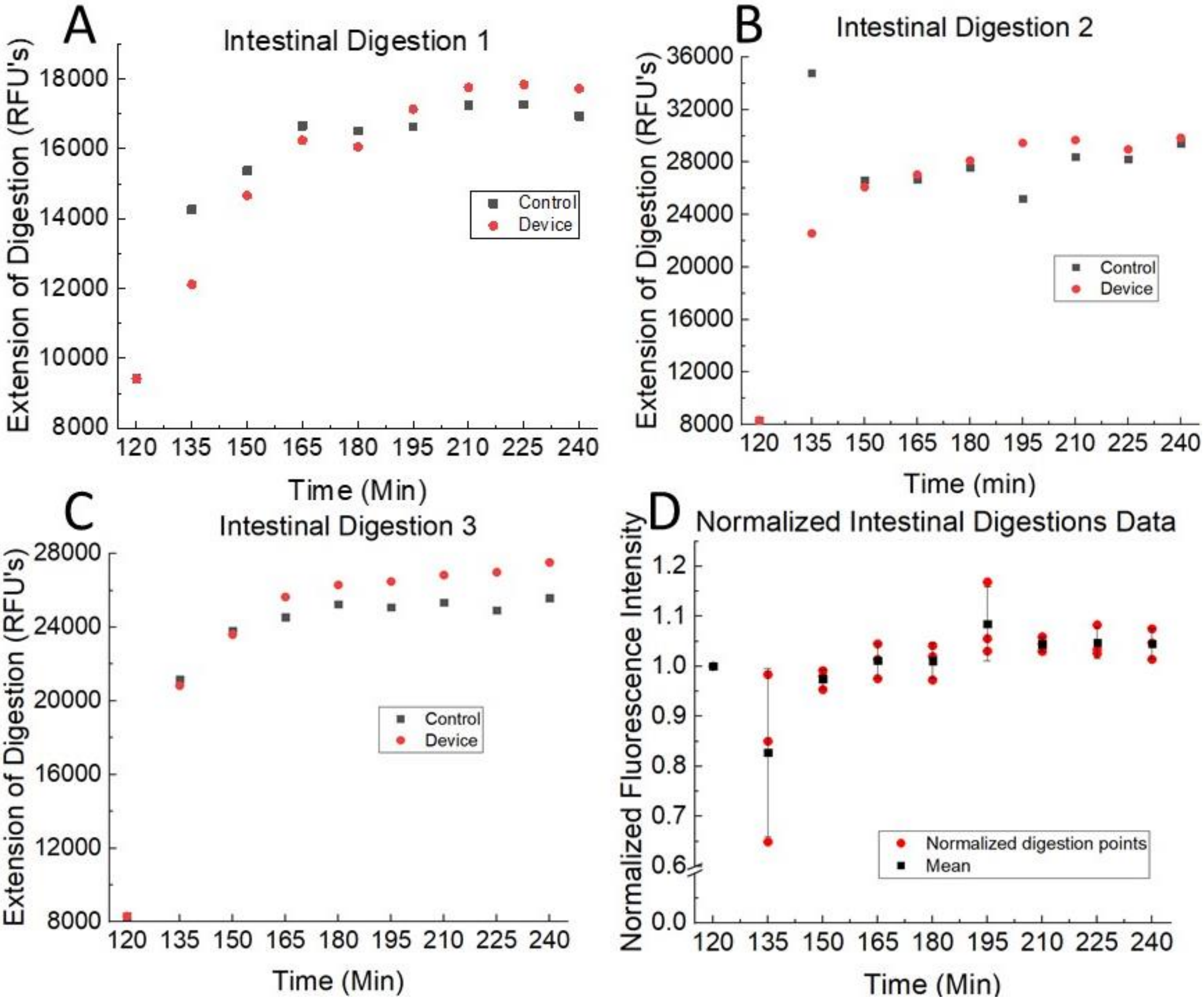


Figure 21 - A, B and C show plots of three intestinal digestions with samples measured at 15-minute intervals over a period of 120 minutes. D is the plot for the normalized data calculated by dividing the results of each digestion by control digestions run simultaneously. The Figure shows Mean +/- STD.

Standalone intestinal digestions were also performed. These were done on and off-chip following gastric digestion by the static INFOGEST protocol. A sample was taken at the end of the gastric digestion to serve as the starting fluorescence value for the intestinal digestion. Samples were measured at 15-minute time points. Again, the normalized data on Figure 21D shows that the mean closely approximates 1 and only showed significant deviation in the first timepoint where the control reading of the experiment 2 (Figure 21B) was identified as an outlier. This could have been due to bubbles affecting the fluorescence reading in the microplate reader. Figure 22 shows an example pH plot of an intestinal digestion. Although there was drift due to old electrodes, the actual pH reading was not affected and was verified by measuring using an external commercial probe at the end of the experiment.

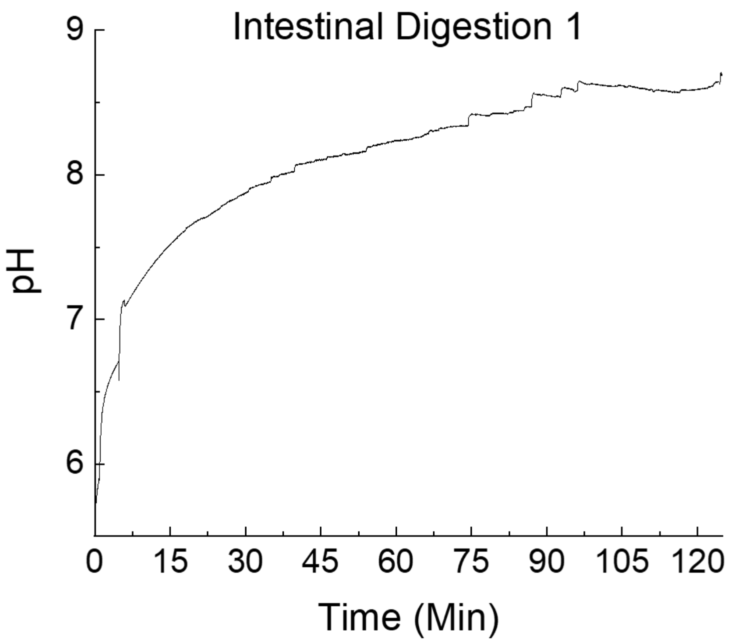


Figure 22 - pH plot for the first intestinal digestion performed. In this digestion experiment, it is possible to see pH adjustment in the first minutes of the experiment and then, a constant drift was observed that was not caused by pH adjustment.

4.9. **Full digestion (gastric+intestinal)**

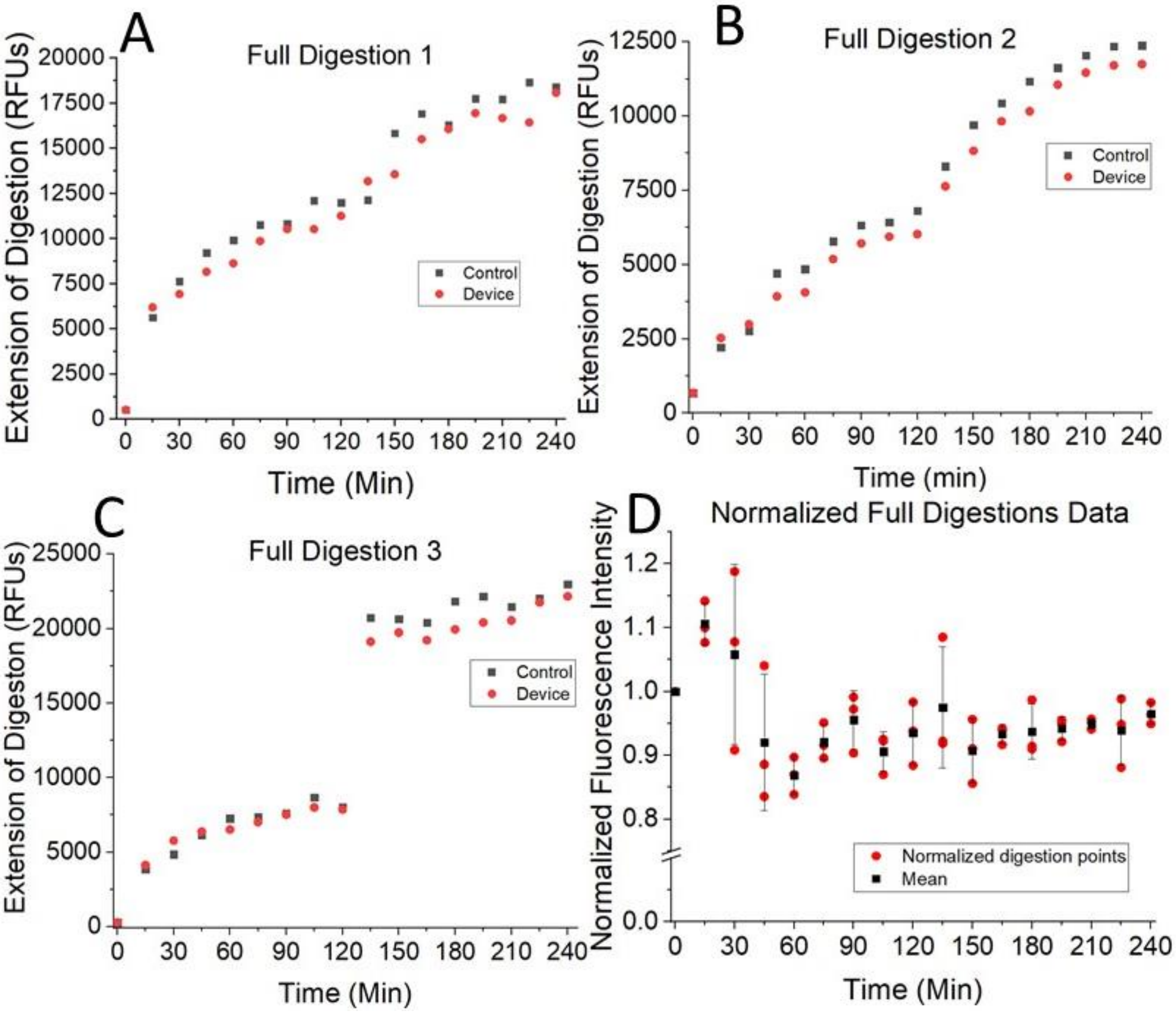


Figure 23 - A, B and C show plots of three full digestion experiments with samples measured every 15 minutes over a period of 120 minutes. D is the plot for the normalized data calculated by dividing the results of each digestion by control digestions run simultaneously. The Figure shows Mean +/- STD.

The full digestion experiments consist of a gastric digestion followed by an intestinal digestion using the output from the first one with the addition of fluid simulants and enzymes for the latter. This experiment was made by collecting liquid from the first phase and adding it to the second manually which will be replaced with the peristaltic pump for gastric emptying in the future. The experiments showed a close approximation to the control experiments. These experiments are more intricate as they run for the longest time and use two pairs of pH electrodes to control each of the chambers separately. Figure 24 shows the combination of the two pH plots traced during the full digestion experiment. From 130 to 140 minutes there was a significant introduction of noise due to the transfer of the digestion fluids between chambers leaving the electrodes temporarily dry for accurate readings.

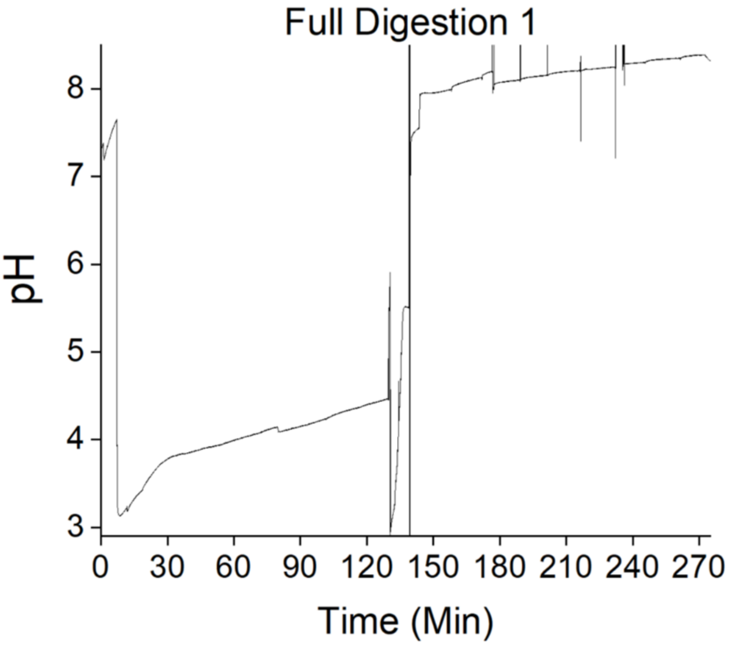


Figure 24 - Example pH plot of a full digestion. The plot is compiled from data of the sensors in the two chambers. The first 125 minutes correspond to the gastric digestion in chamber 1, and the second 125 minutes to the intestinal digestion in chamber 2. Here, the two pH adjustment phases can be seen. The second adjustment phase is distorted due to liquid being transferred and the sensor plot changing from sensor one to two.

These results demonstrate the ability of the developed system to successfully replicate the results from the gold standard INFOGEST protocol. Furthermore, this platform enables an automated real-time control of the two most important parameters affecting the digestion process, pH and temperature, while also reducing the complexity of the experimental setup. In addition, the small amounts of sample and reagents needed for this system open the possibility to evaluate the bioaccessibility of expensive and complex compounds and formulations. Finally, these devices will allow an easy integration of dynamic features of digestions, such as gastric emptying and gradual acidification, which will therefore provide more physiologically relevant results.

5. CONCLUSIONS AND FUTURE PERSPECTIVES

The semi-automated miniaturized platform for human digestion simulation developed and optimized within this work showed robust and reproducible results for both gastric and intestinal phases. Critically, the results obtained were identical to those from the control experiments following the gold standard protocol standardized by the international INFOGEST COST action. The platform was able to efficiently control temperature and pH throughout the experiments without the need of any user input. This represents a major advantage over static methods as it opens the possibility for running semi-dynamic digestion experiments including important digestion processes such as gastric emptying. In addition, real-time pH titrations can also be used as indicators of lipid digestibility.

Improvements to the system may include i) the use of a larger heatsink as the heating element was sometimes limited due to regulator overheating, ii) a PCB for power distribution and Arduino connections so that the system is easier to use and is less likely to short-circuit and shutdown, and iii) either the inclusion of temperature display in MATLAB or re-approaching the sensor PCB encapsulation to allow for direct readings from the data acquisition box and feedback control from MATLAB.

Future work should also make use of the peristaltic pump for experiments including gastric emptying. This will require the optimization of a protocol defining how the emptying rate will be calculated (via volume/calories/other) and, if the simulant fluids will be present in the second chamber or added along with the product of the gastric digestion (which would add another level of complexity).

Finally, device parallelization would represent a significant improvement for higher throughput and to mitigate experimental error. The chips developed and the control system can be replicated if the Arduino pin usage is optimized or more Arduinos are used. Modifications would also be needed in the data acquisition box to connect to a higher number of pH sensors.

Overall, this work achieved a semi-automated and miniaturized device for human intestinal simulations. The device was validated against gold standard protocols and showed the potential to be used in laboratory settings for in vitro testing of compounds aiming for oral administration using reduced volumes and with minimal operator input.

REFERENCES

[1] M. Minekus *et al.*, “A standardised static *in vitro* digestion method suitable for food – an international consensus,” *Food Funct.*, vol. 5, no. 6, pp. 1113–1124, 2014, doi: 10.1039/C3FO60702J.

[2] A. I. Mulet-Cabero *et al.*, “A standardised semi-dynamic: in vitro digestion method suitable for food-an international consensus,” *Food and Function*, vol. 11, no. 2, pp. 1702–1720, Feb. 2020, doi: 10.1039/c9fo01293a.

[3] M. Minekus, P. Marteau, R. Havenaar, and J. H. J. H. in't Veld, “A Multicompartmental Dynamic Computer-controlled Model Simulating the Stomach and Small Intestine:,” *https://doi.org/10.1177/026119299502300205*, vol. 23, no. 2, pp. 197–209, Aug. 2020, doi: 10.1177/026119299502300205.

[4] K. Molly, M. vande Woestyne, and W. Verstraete, “Development of a 5-step multi-chamber reactor as a simulation of the human intestinal microbial ecosystem,” *Applied Microbiology and Biotechnology*, vol. 39, no. 2, pp. 254–258, May 1993, doi: 10.1007/BF00228615.

[5] S. J. Hur, B. O. Lim, E. A. Decker, and D. J. McClements, “In vitro human digestion models for food applications,” *Food Chemistry*, vol. 125, no. 1, pp. 1–12, Mar. 2011, doi: 10.1016/J.FOODCHEM.2010.08.036.

[6] S. Hollebeeck, F. Borlon, Y. J. Schneider, Y. Larondelle, and H. Rogez, “Development of a standardised human in vitro digestion protocol based on macronutrient digestion using response surface methodology,” *Food Chemistry*, vol. 138, no. 2–3, pp. 1936–1944, Jun. 2013, doi: 10.1016/j.foodchem.2012.11.041.

[7] A. G. Oomen, C. J. M. Rompelberg, M. A. Bruil, C. J. G. Dobbe, D. P. K. H. Pereboom, and A. J. A. M. Sips, “Development of an in vitro digestion model for estimating the bioaccessibility of soil contaminants,” *Archives of Environmental Contamination and Toxicology*, vol. 44, no. 3, pp. 281–287, Apr. 2003, doi: 10.1007/s00244-002-1278-0.

[8] I. Alexandropoulou, M. Komaitis, and M. Kapsokefalou, “Effects of iron, ascorbate, meat and casein on the antioxidant capacity of green tea under conditions of in vitro digestion,” *Food Chemistry*, vol. 94, no. 3, pp. 359–365, Feb. 2006, doi: 10.1016/j.foodchem.2004.11.023.

[9] V. Lebet, E. Arrigoni, and R. Amado, “Digestion Procedure Using Mammalian Enzymes to Obtain Substrates forIn VitroFermentation Studies,” *LWT - Food Science and Technology*, vol. 31, no. 6, pp. 509–515, Sep. 1998, doi: 10.1006/FSTL.1998.0402.

[10] A. G. Oomen, C. J. M. Rompelberg, M. A. Bruil, C. J. G. Dobbe, D. P. K. H. Pereboom, and A. J. A. M. Sips, “Development of an in vitro digestion model for estimating the bioaccessibility of soil contaminants,” *Archives of Environmental Contamination and Toxicology*, vol. 44, no. 3, pp. 281–287, Apr. 2003, doi: 10.1007/S00244-002-1278-0.

[11] A. Brodkorb *et al.*, “INFOGEST static in vitro simulation of gastrointestinal food digestion,” *Nature Protocols*, vol. 14, no. 4, pp. 991–1014, Apr. 2019, doi: 10.1038/s41596-018-0119-1.

[12] L. Egger *et al.*, “The harmonized INFOGEST in vitro digestion method: From knowledge to action,” *Food Research International*, vol. 88, pp. 217–225, Oct. 2016, doi: 10.1016/j.foodres.2015.12.006.

[13] T. Bohn *et al.*, “Correlation between in vitro and in vivo data on food digestion. What can we predict with static in _ Enhanced Reader,” *Critical Reviews in Food Science and Nutrition*, vol. 58, no. 13, pp. 2239–2261, Sep. 2018, doi: 10.1080/10408398.2017.1315362.

[14] D. Dupont *et al.*, “Can dynamic in vitro digestion systems mimic the physiological reality?,” *Critical Reviews in Food Science and Nutrition*, vol. 59, no. 10. Taylor and Francis Inc., pp. 1546–1562, May 31, 2019. doi: 10.1080/10408398.2017.1421900.

[15] D. Dupont and A. R. Mackie, “Static and dynamic in vitro digestion models to study protein stability in the gastrointestinal tract,” *Drug Discovery Today: Disease Models*, vol. 17–18. Elsevier Ltd, pp. 23–27, Feb. 04, 2015. doi: 10.1016/j.ddmod.2016.06.002.

[16] O. Ménard *et al.*, “Validation of a new in vitro dynamic system to simulate infant digestion,” *Food Chemistry*, vol. 145, pp. 1039–1045, 2014, doi: 10.1016/j.foodchem.2013.09.036.

[17] K. Molly, M. vande Woestyne, I. de Smet, and W. Verstraete, “Validation of the simulator of the human intestinal microbial ecosystem (SHIME) reactor using microorganism-associated activities,” *Microbial Ecology in Health and Disease*, vol. 7, no. 4, pp. 191–200, 1994, doi: 10.3109/08910609409141354.

[18] C. Li, W. Yu, P. Wu, and X. D. Chen, “Current in vitro digestion systems for understanding food digestion in human upper gastrointestinal tract,” *Trends in Food Science and Technology*, vol. 96. Elsevier Ltd, pp. 114–126, Feb. 01, 2020. doi: 10.1016/j.tifs.2019.12.015.

[19] A. Acevedo-Fani, A. Ochoa-Grimaldo, S. M. Loveday, and H. Singh, “Digestive dynamics of yoghurt structure impacting the release and bioaccessibility of the flavonoid rutin,” *Food Hydrocolloids*, vol. 111, Feb. 2021, doi: 10.1016/j.foodhyd.2020.106215.

[20] S. Iqbal *et al.*, “Evolutions of rheology, microstructure and digestibility of parboiled rice during simulated semi-dynamic gastrointestinal digestion,” *LWT*, vol. 148, Aug. 2021, doi: 10.1016/j.lwt.2021.111700.

[21] O. Ménard *et al.*, “Gastric emptying and dynamic in vitro digestion of drinkable yogurts: Effect of viscosity and composition,” *Nutrients*, vol. 10, no. 9, Sep. 2018, doi: 10.3390/nu10091308.

[22] T. T. Lambers, S. Broeren, J. Heck, M. Bragt, and T. Huppertz, “Processing affects beta-casomorphin peptide formation during simulated gastrointestinal digestion in both A1 and A2 milk,” *International Dairy Journal*, vol. 121, Oct. 2021, doi: 10.1016/j.idairyj.2021.105099.

[23] P. de Haan *et al.*, “Digestion-on-a-chip: A continuous-flow modular microsystem recreating enzymatic digestion in the gastrointestinal tract,” *Lab on a Chip*, vol. 19, no. 9, pp. 1599–1609, May 2019, doi: 10.1039/c8lc01080c.

[24] D. W. Piper and B. H. Fenton, “pH stability and activity curves of pepsin with special reference to their clinical importance.,” *Gut*, vol. 6, no. 5, pp. 506–508, 1965, doi: 10.1136/gut.6.5.506.

[25] S. R. Tello-Solís and B. Romero-García, “Thermal denaturation of porcine pepsin: a study by circular dichroism,” *International Journal of Biological Macromolecules*, vol. 28, no. 2, pp. 129–133, Jan. 2001, doi: 10.1016/S0141-8130(00)00154-9.

[26] A. Agnihotri, “Human Body Respiration Measurement Using Digital Temperature Sensor with I2C Interface,” 2013.

[27] X. J. Yue, “Design of temperature sensor on LM75,” in *5th International Conference on MEMS NANO, and Smart Systems, ICMENS 2009*, 2009, pp. 245–248. doi: 10.1109/ICMENS.2009.61.

[28] M. Gęca, T. Lizak, A. Kociubiński, M. Borecki, and M. L. Korwin-Pawlowski, “Nichrome microheaters as actuators for microfluidic sensors,” in *Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments 2016*, Sep. 2016, vol. 10031, p. 1003126. doi: 10.1117/12.2249184.

[29] C. Knospe, “PID control,” *IEEE Control Systems*, vol. 26, no. 1, pp. 30–31, 2006, doi: 10.1109/MCS.2006.1580151.

[30] S. Atabakhsh, Z. Latifi Namin, and S. Jafarabadi Ashtiani, “Paper-based resistive heater with accurate closed-loop temperature control for microfluidics paper-based analytical devices,” *Microsystem Technologies*, vol. 24, no. 9, pp. 3915–3924, Sep. 2018, doi: 10.1007/s00542-018-3891-5.

[31] M. Yuqing, C. Jianrong, and F. Keming, “New technology for the detection of pH,” *Journal of Biochemical and Biophysical Methods*, vol. 63, no. 1. pp. 1–9, Apr. 29, 2005. doi: 10.1016/j.jbbm.2005.02.001.

[32] D. J. Graham, B. Jaselskis, and C. E. Moore, “Development of the glass electrode and the pH response,” *Journal of Chemical Education*, vol. 90, no. 3, pp. 345–351, Mar. 2013, doi: 10.1021/ed300246x.

[33] R. H. G. Mingels, S. Kalsi, Y. Cheong, and H. Morgan, “Iridium and Ruthenium oxide miniature pH sensors: Long-term performance,” *Sensors and Actuators B: Chemical*, vol. 297, p. 126779, Oct. 2019, doi: 10.1016/j.snb.2019.126779.

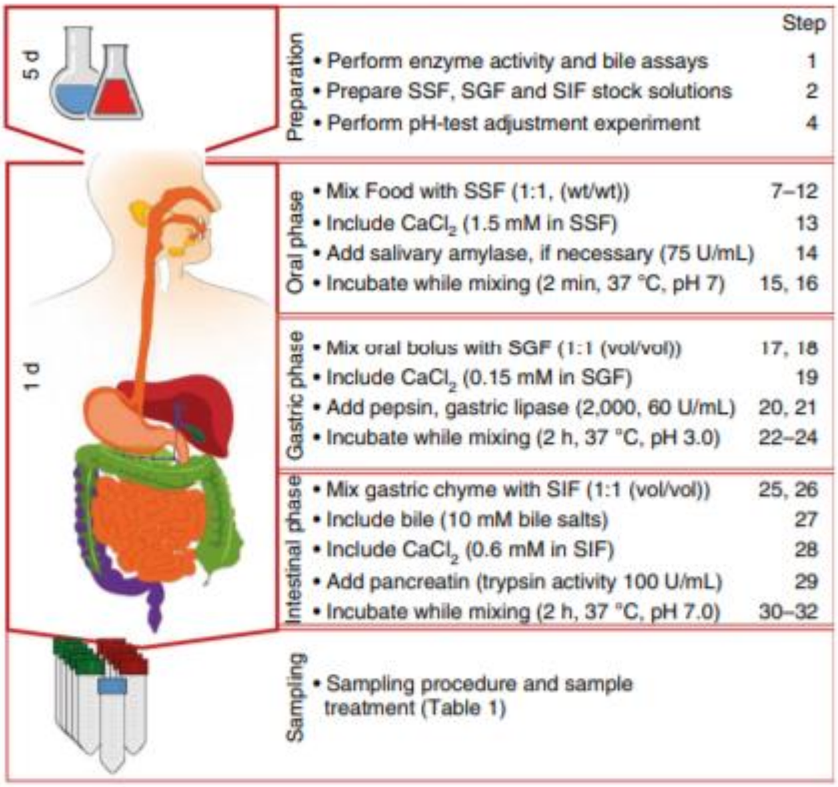
[34] J. Lee, H. K. Musyimi, S. A. Soper, and K. K. Murraya, “Development of an Automated Digestion and Droplet Deposition Microfluidic Chip for MALDI-TOF MS,” 2008, doi: 10.1016/j.jasrns.2008.03.015.

[35] J. E. Sosa-Hernández *et al.*, “Organs-on-a-Chip Module: A Review from the Development and Applications Perspective,” *Micromachines*, vol. 9, no. 10, p. 536, Oct. 2018, doi: 10.3390/mi9100536.

[36] A. Volpe, U. Krishnan, M. S. Chiriacò, E. Primiceri, A. Ancona, and F. Ferrara, “A Smart Procedure for the Femtosecond Laser-Based Fabrication of a Polymeric Lab-on-a-Chip for Capturing Tumor Cell,” *Engineering*, Dec. 2020, doi: 10.1016/j.eng.2020.10.012.

[37] M. W. Toepke and D. J. Beebe, “PDMS absorption of small molecules and consequences in microfluidic applications,” *Lab on a Chip*, vol. 6, no. 12. Royal Society of Chemistry, pp. 1484–1486, 2006. doi: 10.1039/b612140c.

A. INFOGEST protocols

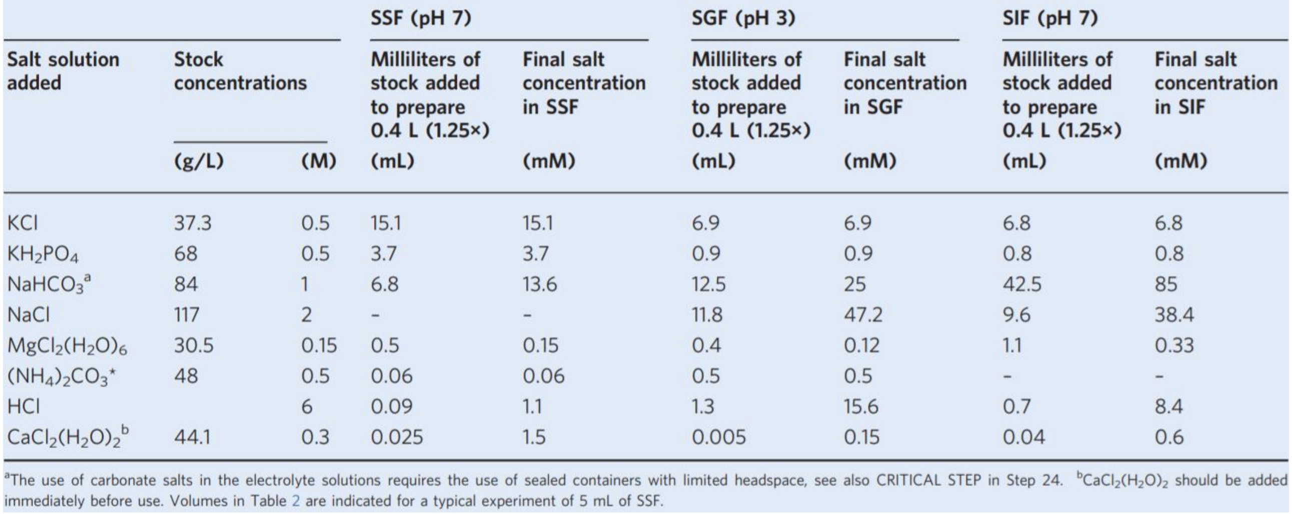


	Step
• Perform enzyme activity and bile assays	1
• Prepare SSF, SGF and SIF stock solutions	2
• Perform pH-test adjustment experiment	4
• Mix Food with SSF (1:1, (wt/wt))	7–12
• Include $CaCl_2$ (1.5 mM in SSF)	13
• Add salivary amylase, if necessary (75 U/mL)	14
• Incubate while mixing (2 min, 37 °C, pH 7)	15, 16
• Mix oral bolus with SGF (1:1 (vol/vol))	17, 18
• Include $CaCl_2$ (0.15 mM in SGF)	19
• Add pepsin, gastric lipase (2,000, 60 U/mL)	20, 21
• Incubate while mixing (2 h, 37 °C, pH 3.0)	22–24
• Mix gastric chyme with SIF (1:1 (vol/vol))	25, 26
• Include bile (10 mM bile salts)	27
• Include $CaCl_2$ (0.6 mM in SIF)	28
• Add pancreatin (trypsin activity 100 U/mL)	29
• Incubate while mixing (2 h, 37 °C, pH 7.0)	30–32
• Sampling procedure and sample treatment (Table 1)	

Figure 25 - INFOGEST protocol adapted from [1]

In vitro and on-chip gastro-intestinal digestions were carried out following the standardized INFOGEST COST action protocol first published [1] in 2014 and later updated in 2019 [11]. The composition of the simulated digestion fluids can be found in Figure 24.

Briefly, for digestions, oral incubation phase was skipped because it did not play a role in the quenched casein digestion. Samples from oral phase and gastric phase were mixed 1:1 and inserted into the control 15mL centrifuge tube incubated out in a thermomixer (Eppendorf™, Hamburg, Germany) or inserted inside the digestion device produced. After the 2h incubation, the result was mixed 1:1 with the intestinal phase and the reaction allowed to occur for further 2h. Samples were collected every 15 minutes with a volume of 100 uL and placed in a 96 well plate to read.



			SSF (pH 7)		SGF (pH 3)		SIF (pH 7)	
Salt solution added	Stock concentrations		Milliliters of stock added to prepare 0.4 L (1.25×)	Final salt concentration in SSF	Milliliters of stock added to prepare 0.4 L (1.25×)	Final salt concentration in SGF	Milliliters of stock added to prepare 0.4 L (1.25×)	Final salt concentration in SIF
	(g/L)	(M)	(mL)	(mM)	(mL)	(mM)	(mL)	(mM)
KCl	37.3	0.5	15.1	15.1	6.9	6.9	6.8	6.8
KH_2PO_4	68	0.5	3.7	3.7	0.9	0.9	0.8	0.8
$NaHCO_3$[a]	84	1	6.8	13.6	12.5	25	42.5	85
NaCl	117	2	-	-	11.8	47.2	9.6	38.4
$MgCl_2(H_2O)_6$	30.5	0.15	0.5	0.15	0.4	0.12	1.1	0.33
$(NH_4)_2CO_3$*	48	0.5	0.06	0.06	0.5	0.5	-	-
HCl		6	0.09	1.1	1.3	15.6	0.7	8.4
$CaCl_2(H_2O)_2$[b]	44.1	0.3	0.025	1.5	0.005	0.15	0.04	0.6

[a]The use of carbonate salts in the electrolyte solutions requires the use of sealed containers with limited headspace, see also CRITICAL STEP in Step 24. [b]$CaCl_2(H_2O)_2$ should be added immediately before use. Volumes in Table 2 are indicated for a typical experiment of 5 mL of SSF.

Figure 26 - Protocol for simulant fluids adapted from the INFOGEST protocol [1]. These are the volumes required to produce 400 mL of sample diluted with water. These fluids were kept in a flask at 1º C and aliquots were taken whenever need for an experiment.

B. Digestion protocol for control and device digestion

The following protocol shows the procedure for a complete digestion on the device produced along with the control. Volume is 2mL for both experiments

Conditions:

•10 µg/mL substrate + enzymes (N=1)

Preparation:

Day -1:

•Weigh enzymes (~13.84mg pepsin in 1000µLMilli-Q water; ~80mg bile in 1000µL in SIF; ~370.38mg pancreatin in 2500µL SIF).

Day 0:

•Dissolve the enzymes (pepsin in H2O 13.84 mg/mL; bile and pancreatin in SIF 80mg/mL and 148.15 mg/mL respectively)

Note: Pancreatin should be dissolved on ice. Agitate with a magnet for about 10’, centrifuge at 620g for another 10’ at 4 ºC and use supernatant.

•Prepare 1x digestion buffer (1200µL: 60µL 20x + 1140µL dH_2O)

•Prepare 1100µL of substrate at 10 µg/mL [(11µL stock solution (1mg/mL) + 1089 µL digestion buffer)]

•Prepare SSF:960µL SSF + 234µL H2O + 6µL $CaCl_2$

•Prepare SGF + pepsin: 2000µLSGF+ 1.25µL $CaCl_2$+ 7.5 µL HCl + 391µL H2O

From the previous solution collect 900µL and add to the sample + SSF and add 100 µL pepsin

•Prepare SIF + bile: 1925µL SIF + 825µL bile + 1.25µL $CaCl_2$

From the previous solution collect 550.66µLand add to the sample + SSF + SGF and add 4.4µLNaOH +194.66µL H_2O+ 250µL pancreatin

C. pH probes fabrication protocol

The following protocol shows how the pH probes were made, reference and main electrode.

The protocol changes slightly as miniaturized versions of the reference electrode were built using 10uL pipette tips.

Reference electrode fabrication – Ag/AgCl electrode in 3M KCl

Materials

- 0.1 M HCl solution
- Sodium alginate solution
- CaCl2 solution (3g/100mL) – to cure the alginate
- Ag wire (wire characteristics)
- Pt wire (50x0.2 mm Sigma Ep1330-1EA)
- Cotton
- 200 µL pipette tip
- 3M KCl

Steps

- Place a little bit of cotton at the end of the pipette tip
- Fill the area with the cotton with 20-30 uL of the alginate solution
- Add ~10uL of CaCl2 solution from the top of the tip
- Dip the bottom of the tip for at least 5 minutes in the CaCl2 solution
- Ag wire coating with AgCl (while waiting for the curing of the alginate)
    - Expose a length of 1.5 cm
    - Place the Pt wire and the Ag wire inside the 0.1M HCl solution
    - Apply a fixed current of 2.355 mA for 1 minute – Ag connected to signal and Pt to ground – the exposed are of the Ag wire must turn black
    - Rinse the wire with DI water
- Fill the tip with 3M KCl – ensure that there is no leakage from the bottom of the tip
- Place the Ag/AgCl wire (only the dark area) inside the 3M KCl solution and seal the top of the tip with parafilm

pH sensing electrode fabrication – IrOx deposition in Ti wire

Materials

- COCKTAIL SOLUTION (for 10 mL of volume)
    - Dissolve 15 mg of Iridium chloride hydrate ($IrCl_4.H_2O$) in 10 mL of DI water (magnetic stir for 30 min)
    - Add 100 µL of aqueous 30% H2O2 and stir 10 min
    - Add 50 mg of oxalic acid dehydrate ($(COOH)_2.2H_2O$) and stir 10 min
    - Adjust the pH of the solution slowly to 10.5 by adding small portions of anhydrous potassium carbonate (K2CO3) – resulting solution must be yellow
    - Cover the solution (to protect it from light) and leave it at room temperature for 2 days to stabilize. When the colour of the solution changes from yellow to light-violet it is ready to be used for deposition
    - The solution may be kept in the dark at 4ºC until is used – it lasts at least a couple of months
- 0.5 M H2SO4 solution
- Pt wire reference (50x0.2 mm Sigma Ep1330-1EA)
- Ti wire for IrOx deposition (0.25 mm diameter Sigma 460400-2.2G)
- Nail polish

Steps

- Use the Nail polish to insulate the Ti wire leaving a length of 1.5 cm exposed
- Clean the electrodes by oxidizing and reducing (Cyclic Voltammetry) by applying a -0.23V to +1.1V at 100mV/s rate for 20 cycles in the H_2SO_4 solution (0.5 M) – connect the Ti wire to the signal and the Pt wire to the ground and use the Keithly 2400 in the MEMS lab with the MATLAB script connected to the computer with the RS-232 cable
- When the MATLAB program is completed, dip both wires in DI water and dry
- With the same connections, apply a constant current of 0.1 mA (specific for the area exposed – 1.5 cm length 0.25 mm diameter Ti wire) for 20 minutes (the voltage reading should be around 2.1 V)
- After the deposition the exposed area of the Ti wire should turn dark blue

Rinse the IrOx electrode with DI water, dry with N2 and store in PBS for 2 days to stabilize. After 2 days the electrode is ready to be calibrated and used

D. Arduino function

This is a full dump of the Arduino code (with Arduino's syntax highlighted) used in the final device both for temperature control as well as pH feedback.

```
#include <Wire.h>
#include "Protocentral_MAX30205.h"

MAX30205 sensor1;
MAX30205 sensor2;

const int chamber1 = 5; // output pwm pin
const int chamber2 = 6; // output pwm pin
float SV = 37; // Set Point
float PV=0;
float error = 0;
float actualTime = 0;
float previousTime = 0;
float Dt = 0;
float out = 0;
float Derror =0;
float previousError=0;
float Kp, Ki, Kd;
float P;
float I;
float D;
int i;
int outPin = chamber1;
float StatePreviousTime;
float temp = 0;

//Codigo pump

#include <AccelStepper.h>
AccelStepper acid(AccelStepper::DRIVER,10,11);
AccelStepper base(AccelStepper::DRIVER,12,13);
int steps1 = 0;
int steps2 = 0;
const byte buffSize=64;
char inputBuffer[buffSize];
const char starter='<';
const char finisher='>';
byte received=0;
boolean reading=false;
boolean newread=false;
char PCmessage[buffSize]={0};
char mode[buffSize]={0};
int pwm1 = 0;
int pwm2 = 0;


void setup() {

  Serial.begin(9600);
  //Temperature sensors
  Wire.begin();
  //Assign sensors addresses
  sensor1.sensorAddress = MAX30205_ADDRESS1;
  sensor2.sensorAddress = MAX30205_ADDRESS2;
  sensor1.begin();
  sensor2.begin();

  acid.setMaxSpeed(10000);
  base.setMaxSpeed(10000);
  acid.setSpeed(10000);
  base.setSpeed(10000);
  acid.setAcceleration(10000);
  base.setAcceleration(10000);

  mode[0] = 'C';

  pinMode(chamber1, OUTPUT);
  pinMode(chamber2, OUTPUT);
  i = 1;
  Kp=30; Ki=0.1; Kd=450; // initial parameters for PID constants
  previousTime = millis()/1000.0;
}

void PID(){
  actualTime = millis()/1000.0;
  Dt = actualTime-previousTime;
  previousTime = actualTime;

  error = SV - PV;
  Derror = error-previousError;
  previousError = error;
  P = Kp*error;
  I = I + Ki * error * Dt;
```

```
 I = constrain(I, -10, 50);
 D = (Kd*Derror)/Dt;
 out = P + I + D;
 out = constrain(out, 0, 75);
 analogWrite(outPin, int(out));
}

void getPCData() {

  if (Serial.available()){

  char x=Serial.read();
   if (x==finisher){
    reading=false;
    newread=true;
    inputBuffer[received]=0;
    return parseData();
   }

   if (reading){
    inputBuffer[received]=x;
    received++;
   }

   if (x==starter){
    reading=true;
    received=0;
   }
 }
}

void parseData() {

char* strtokIndex;

//<mode,pwm,pwm2,steps1,steps2>

strtokIndex=strtok(inputBuffer,":");
strcpy(mode, strtokIndex);

strtokIndex=strtok(NULL,":");
String motorstr(strtokIndex);
pwm1=atoi(strtokIndex);


strtokIndex=strtok(NULL,":");
pwm2=atof(strtokIndex);

strtokIndex=strtok(NULL,":");
steps1=atof(strtokIndex);

strtokIndex=strtok(NULL,":");
steps2=atof(strtokIndex);

return;
}

void power() {
acid.move(steps1);
acid.runToPosition();
base.move(steps2);
base.runToPosition();
steps1 = 0;
steps2 = 0;
}

void loop() {

 if (mode[0] == 'A') {

  outPin = chamber1;
  PV = sensor1.getTemperature();
  analogWrite(chamber2, 0);
  getPCData();
  power();
  PID();
  analogWrite(3, 5);
 }
 else if (mode[0] == 'B') {

  outPin = chamber2;
  PV = sensor2.getTemperature();
  analogWrite(chamber1, 0);
  getPCData();
  power();
  PID();
  analogWrite(3, 5);
 }
 else if (mode[0] == 'C') {
  getPCData();
  analogWrite(3, 3);
  analogWrite(outPin, 0);
  analogWrite(chamber1, 0);
  analogWrite(chamber2, 0);
 }
 else if (mode[0] != 'A' && mode[0] != 'B' && mode[0] != 'C') {
  getPCData();
  analogWrite(3, 3);
  analogWrite(chamber1, 0);
  analogWrite(chamber2, 0);
  Serial.print("Chamber 1 :  ");
  Serial.println(sensor1.getTemperature());
  Serial.print("Chamber 2 :  ");
  Serial.println(sensor2.getTemperature());
  Serial.println(mode[0]);
  delay(200);
}
}
```

E. MATLAB main function

This is a full code dump for MATLABs main function (with MATLABs syntax highlighted). This function was used to open serial communication and define some variables for the pH control as well as temperature control which is not used in this version of the device.

```
function res=Start_pH(MyDataName)
% This functions starts the pH logger interface
% It is for 2-ch pH and temperature logger
% Input parameter is string of name for the data e.g. 'MyTest1'
% (C) Alar Ainla, International Iberian Nanotech Lab (INL), 2021
% -------------------------------------------------------------------------
% Please define here the COM/serial port, which is used for the device
serialPortBox='COM5'; % Please define here COM port, to which the interface is connected
serialPortArduino = 'COM4';
% Baudrate 115200, Databits 8, Flow control none, parity none, terminator
% <CR> which corresponds to ascii code 13
% Input parameter is name for the dataset

% -------------------------------------------------------------------------
% CODE

% My global data
global myobj

% --- This has following variables inside ---
% .sobj - serial port
% .j - counter index for data points
% .D1 - Digital signal channel 1 (A) / Digital word from the device
% .V1 - Voltage channel 1 (A) / Calibrated value in mV
% .D2 - Digital signal channel 2 (B) / Digital word from the device
% .V2 - Voltage channel 2 (B) / Calibrated value in mV
% .T1 - Temperature channel 1 (C)
% .T2 - Temperature channel 2 (D)
% .t - time in seconds
% .DT - time vector
% .dataName - data set name

global myfig

% Setup figure
myfig = figure(1);

% Initialize variables
myobj.j=0;
myobj.D1=[];
myobj.D2=[];
myobj.V1=[];
myobj.V2=[];
myobj.T1=[];
myobj.T2=[];
myobj.t=[];
myobj.DT=[];
myobj.Acido=[];
myobj.Base=[];
myobj.dataName=MyDataName;
myobj.t0=TimeInSec(clock);

% Setup com port
myobj.sobj=serial(serialPortBox,'BaudRate',115200,'DataBits',8); % Flow control is none and parity is none
fopen(myobj.sobj);
set(myobj.sobj,'Terminator',13); % Terminator <CR>
set(myobj.sobj,'BytesAvailableFcnMode','terminator');
set(myobj.sobj,'BytesAvailableFcn',{@SerialDataReceived}); % Setup what happens when data comes in



% Return link to the object
res=myobj;

%Code:
myobj.a = serialport(serialPortArduino, 9600);
myobj.previousTime1 = 0.0;
myobj.previousError1 = 0.0;
myobj.previousTime2 = 0.0;
myobj.previousError2 = 0.0;
myobj.I1 = 0.0;
myobj.I2 = 0.0;

myobj.pwm1 = 0;
myobj.pwm2 = 0;

myobj.A = 0;
myobj.maxt1 = 0;
myobj.maxt2 = 0;

%Reta Cal
myobj.n1measure9 = 135.9;
myobj.n1measure7 = 290.1;
myobj.n1measure4 = 514.5;
%myobj.n2measure9 = 0;
%myobj.n2measure7 = 0;
%myobj.n2measure4 = 0;
myobj.n2measure9 = myobj.n1measure9;
myobj.n2measure7 = myobj.n1measure7;
myobj.n2measure4 = myobj.n1measure4;

myobj.x1 = [4 7 9];
myobj.x2 = [4 7 9];
myobj.y1 = [myobj.n1measure4 myobj.n1measure7 myobj.n1measure9];
myobj.y2 = [myobj.n2measure4 myobj.n2measure7 myobj.n2measure9];

myobj.polyfit1 = polyfit(myobj.y1, myobj.x1, 1);
myobj.polyfit2 = polyfit(myobj.y2, myobj.x2, 1);

myobj.cal3 = (1/myobj.polyfit1(1));
myobj.cal7 = (1/myobj.polyfit2(1));
myobj.offset1 = myobj.polyfit1(2);
myobj.offset2 = myobj.polyfit2(2);

myobj.pH3 = (3-myobj.offset1)*myobj.cal3;
myobj.pH7 = (7-myobj.offset1)*myobj.cal7;
myobj.stepsAcido = 0;
myobj.stepsBase = 0;


end
```

F. MATLAB read and control function

This is a full code dump for MATLABs read function (with MATLABs syntax highlighted) which was activated every time data was sent from the data acquisition box. This function also called for the control functions, plotted the data received and sent control information to Arduino as well as displaying information for the user such as time elapsed, current pH, and some other information.

```
function SerialDataReceived(srch,eventdata)
% This is a function which is called every time that data is received from
% the pH sensor interface over Serial port
% This function processes the data, makes plot, makes backup saving etc
% (C) Alar Ainla, International Iberian Nanotech Lab (INL), 2021
% ------------------------------------------------------------------------

% Calibration constants - these are for the potential calibration
% do convert the ADC readings to physical voltage in mV

% OLD VALUES
%V_a=8.009033e-06; % mV/dig - digital calibration constant slope
%V_b=-1.079705e+02; % mV - digital calibration constant offset
% NEW VALUES
V_a=2.456977e-05; % mV/dig - digital calibration constant slope
V_b=-3.140828e+02; % mV - digital calibration constant offset



global myobj
global myfig

if ~(myobj.sobj==0) % Serial port is open
    if(get(myobj.sobj,'BytesAvailable')>0) % Data has come
        datain=fscanf(myobj.sobj);
        % Process the incoming data to value A, B, C, D
        values_str=split(datain,"A");
        values_str=split(values_str(2),",B");
        A=str2double(values_str(1));
        values_str=split(values_str(2),",C");
        B=str2double(values_str(1));
        values_str=split(values_str(2),",D");
        C=str2double(values_str(1));
        values_str=split(values_str(2),";");
        D=str2double(values_str(1));
        % Store values
        myobj.j=myobj.j+1;
        myobj.D1(myobj.j)=A;
        myobj.D2(myobj.j)=B;
        myobj.T1(myobj.j)=round(C,2);
        myobj.T2(myobj.j)=round(D,2);
        myobj.DT{myobj.j}=clock;
        myobj.t(myobj.j)=TimeInSec(myobj.DT{myobj.j})-myobj.t0;
        % Compute voltages
        myobj.V1(myobj.j)=round((myobj.D1(myobj.j)*V_a+V_b),1);
        myobj.V2(myobj.j)=round((myobj.D2(myobj.j)*V_a+V_b),1);
        % Make plots
        subplot(1,2,1)
        plot(myobj.t,myobj.V1,'r');
        hold on
        plot(myobj.t,myobj.V2,'b');
        xlabel 'Time (s)'
        ylabel 'Electrode potential (mV)'
        title(['V1: ' num2str(myobj.V1(myobj.j)) 'mV, V2: ' num2str(myobj.V2(myobj.j)) 'mV']);
        hold off
        subplot(1,2,2)
        plot(myobj.t,myobj.T1,'r');
        hold on
        plot(myobj.t,myobj.T2,'b');
        xlabel 'Time (s)'
        ylabel 'Temperature (C)'
        title(['T1: ' num2str(myobj.T1(myobj.j)) 'C, T2: ' num2str(myobj.T2(myobj.j)) 'C']);
        hold off

        % Save data - backup
        if mod(myobj.j,1000)==0
            save([myobj.dataName '.mat'],'myobj');
        end
        %Code
        steps1 = 0;
        steps2 = 0;
        if (myobj.t(length(myobj.t))/60) > 2
            if mod(myobj.j,40)==0
                if (myobj.t(length(myobj.t))/60) > 130
                    steps2 = StepsBase(round(myobj.V2(myobj.j)));
                    myobj.stepsBase = myobj.stepsBase + steps2;
                    myobj.Base(myobj.j) = steps2;
                else
                    steps1 = StepsAcido(round(myobj.V1(myobj.j)));
                    myobj.stepsAcido = myobj.stepsAcido + steps1;
                    myobj.Acido(myobj.j) = steps1;
                end
            end
        end
        if mod(myobj.j,2)==0
            if (myobj.t(length(myobj.t))/60) > 130 %130
                [myobj.pwm2,myobj.I2,myobj.previousTime2,myobj.previousError2] = PIDTemp(D,myobj.j,myobj.I2,myobj.previousTime2,myobj.previousError2,2);
            else
                [myobj.pwm1,myobj.I1,myobj.previousTime1,myobj.previousError1] = PIDTemp(C,myobj.j,myobj.I1,myobj.previousTime1,myobj.previousError1,1);
            end
        end

        if (myobj.t(length(myobj.t))/60) > 130
```

```
            writeline(myobj.a,['<B:',num2str(myobj.pwm1),':',num2str(myobj.pwm2),':',num2str(steps1),':',num2str(steps2),'>']);
        else
            writeline(myobj.a,['<A:',num2str(myobj.pwm1),':',num2str(myobj.pwm2),':',num2str(steps1),':',num2str(steps2),'>']);
        end

        if myobj.maxt1 < C
            myobj.maxt1 = C;
        end
        if myobj.maxt2 < D
            myobj.maxt2 = D;
        end

        %Check Output:
        if (myobj.t(length(myobj.t))/60) > 130
            fprintf(['<B:',num2str(myobj.pwm1),':',num2str(myobj.pwm2),':',num2str(steps1),':',num2str(steps2),'>',' StepsÃ?cido:',num2str(myobj.stepsAcido),' StepsBase:',num2str(myobj.stepsBase),' uL Ã?cido:',num2str((myobj.stepsAcido)*(0.00089 )),' uL Base:',num2str((myobj.stepsBase)*(0.04653211/16))]);
        else
            fprintf(['<A:',num2str(myobj.pwm1),':',num2str(myobj.pwm2),':',num2str(steps1),':',num2str(steps2),'>',' StepsÃ?cido:',num2str(myobj.stepsAcido),' Steps-Base:',num2str(myobj.stepsBase),' uL Ã?cido:',num2str((myobj.stepsAcido)*(0.00089 )),' uL Base:',num2str((myobj.stepsBase)*(0.04653211/16))]);
        end
        fprintf('\n');
        fprintf([char(duration(0,0,myobj.t(length(myobj.t)))),' pH1 : ', num2str(round((myobj.V1(myobj.j)/myobj.cal3),1) + myobj.offset1), ' pH2 : ', num2str(round((myobj.V2(myobj.j)/myobj.cal7),1) + myobj.offset2), ' Temp max1 : ', num2str(round(myobj.maxt1,2)), ' Temp max2 : ', num2str(round(myobj.maxt2,2))]);
        fprintf('\n');
    end
end
end
```

G. MATLAB control functions (pH and temperature)

Chamber 1 pH control:

This is a full code dump for MATLABs chamber 1 pH control (with MATLABs syntax highlighted).

This function was used to calculate the difference in mV from the desired setpoint for the first device chamber (Gastric chamber). The setpoint is the mV for pH desired according to the calibration previously made and the further the read mV are from the desired setpoint, the more HCl is added.

```
function steps = StepsAcido(mV)

global myobj

desmV = myobj.pH3;
dif = desmV-mV;

if dif > 300

    steps = 1800;

elseif dif > 200

    steps = 1300;

elseif dif > 100

```

```
    steps = 700;

elseif dif > 50

    steps = 200;

elseif dif > 15

    steps = 100;

else

    steps = 0;

end
end
```

Chamber 2 pH control:

This is a full code dump for MATLABs chamber 2 pH control (with MATLABs syntax highlighted).

This function was used to calculate the difference in mV from the desired setpoint for the second device chamber (Intestinal chamber). The setpoint is the mV for pH desired according to the calibration previously made and the further the read mV are from the desired setpoint, the more NaOH is added.

```
function steps = StepsBase(mV)

global myobj

desmV = myobj.pH7;

dif = mV-desmV;

if dif > 300

    steps = 1800;

elseif dif > 200

    steps = 1300;

elseif dif > 100

    steps = 700;

elseif dif > 50

    steps = 200;

elseif dif > 10

    steps = 100;

else

    steps = 0;

end
end
```

PID temperature control for both chambers (unused in this version):

This was the MATLAB code developed for temperature control when the sensors were still read via data acquisition box. Although the code is not used, it is the goal of the project to use it in the future so it is included in the appendix. (MATLABs syntax highlighting was used)

```
function [pwm,I,previousTime,previousError] = PIDTemp(PV,t,I,previousTime,previousError,chamber)

%settings
SV = 37.0;

if chamber == 1
    if t < 2000
        Kp = 30.0;
        Ki = 0.3;
        Kd = 5000.0;
    elseif t < 2300
        Kp = 15.0;
        Ki = 0.1;
        Kd = 1000.0;
    else
        Kp = 5.0;
        Ki = 0.05;
        Kd = 0;
    end
elseif chamber == 2
    if t < 27200
        Kp = 30.0;
        Ki = 0.3;
        Kd = 5000.0;
    elseif t <= 27500
        Kp = 15.0;
        Ki = 0.1;
        Kd = 1000.0;
    elseif t > 27500
        Kp = 5.0;
        Ki = 0.05;
        Kd = 0;
    end
end

%Nichrome:
%P: 25
%I: 0.1
%D: 0.1

%Time
actualTime = t;
Dt = actualTime - previousTime;
previousTime = actualTime;

%Error
error = SV - PV;
Derror = error - previousError;
previousError = error;

%PID
P = Kp*error;
I = I + Ki*(error*Dt);
I = min(max(I, 0), 50);
D = Kd*(Derror/Dt);
pwm = P+I+D;

%output
pwm = round(min(max(pwm, 0), 255),0);
pwm = min(pwm,125);
pwm = min(pwm,100);
pwm = min(pwm,75);
%pwm = min(pwm,50);

end
```

2021

Tiago Nuno Gomes Dias

Development of an automated microfluidics-based device for simulating human gastrointestinal digestion

NOVA SCHOOL OF SCIENCE & TECHNOLOGY